Changing the Way the World Thinks about Computer Security

Sarah Gordon Ford

School of Computing Science

Middlesex University

London, UK

Context document, submitted in partial fulfillment for the degree of Doctor of Philosophy

by publication

August 2004

Revisions submitted January 13, 2006

Signatu

CHANGING THE WAY THE WORLD THINKS ABOUT COMPUTER SECURITY 1

Acknowledgements and Dedication 4

Preface 5

Introduction 8

General Overview 10

Comparability: Quality, Scale and Scope 15

Research Context 18

Technology 19

Non-Technical Issues 20

Research Submission Overview 32

Predicting our Future: Technologically Enabled Crime 34

Virus Writers – Unraveling the Mystery 36

The First Macro Viruses: Concept and Excel 41

A New Concept 41

End of the Innocence: Changes in the Virus Writing Population? 45

A Holistic Strategy for Virus Prevention 49

Toward More Meaningful Tests of Antivirus Software 50

Cyberterrorism: Fact or Fiction? 54

Privacy: Do As I Say, Not As I Do 56

Summary of Data Gathering Issues 59

Reflections on Challenges 61

Summary of Research Progress and Impact 64

Topics for Future Research **66**

Conclusion **69**

References **73**

Appendix 1: List of work submitted as part of this application 84

Appendix 2: Quotes related to this research 87

Appendix 3: Television, radio and magazine appearances as expert on social and technical aspects of computer security. 91

Diagram 2 (Enlarged) 98

Acknowledgements and Dedication

For encouragement, support and much guidance, without which this publication would not be possible, I would like to thank Dr. Colin Tully, Dr. Mark Woodman and Dr. Penny Duquenoy; special thanks to Dr. Claudia Kalay for remote assistance.

Embarking in this research area would have been even more difficult if not for people like Dr. Louise Yngstrom, to whom I owe a dept of gratitude for encouragement both academic and personal.

Others helped me along the way, too many to mention, but several I really must thank individually: The first person to ever tell me I actually would change the world with my research was Tom Wachtel; I would not be writing this if not for both his insight and help. Others include Jon, Tim and Padgett; Jack, Larry, Stevie,. Thanks, guys.

I also thankfully acknowledge my mentor, the late Dr. Harold Highland, who told me I not only could, but must do this research and publish it.

Finally, I would like to thank Dr. Richard A. Ford, without whom none of this would matter. I somewhat self-indulgently dedicate this thesis to him, borrowing from the lyricism of T. S. Eliot:

Dedicated to Richard ....."to whom I owe the leaping delight
That quickens my senses in our wakingtime .....
No peevish winter wind shall chill
No sullen tropic sun shall wither
The roses in the rose-garden which is ours and ours only..
But this dedication is for others to read:
These are private words addressed to you in public."

Thank you.

Preface

Small changes in an established system can result in larger changes in the overall system (e.g. network effects, emergence, criticality, broken windows theory). However, in an immature discipline, such as computer security, such changes can be difficult to envision and even more difficult to implement, as the immature discipline is likely to lack the scientific framework that would allow for the introduction of even minute changes. (Cairns, P. and Thimbleby, H, 2003) describe three of the signs of an immature discipline as postulated by (Kuhn, 1970):

a. squabbles over what are legitimate tools for research
b. disagreement over which phenomenon are legitimate to study, and
c. inability to scope the domain of study.

The research presented in this document demonstrates how the computer security field, at the time this research began, was the embodiment of these characteristics. It presents a cohesive analysis of the intentional introduction of a series of small changes chosen to aid in maturation of the discipline. Summarily, it builds upon existing theory, exploring the combined effect of coordinated and strategic changes in an immature system and establishing a scientific framework by which the impact of the changes can be quantified.

By critically examining the nature of the computer security system overall, this work establishes the need for both increased scientific rigor, and a multidisciplinary approach to the global computer security problem. In order for these changes to take place, many common assumptions related to computer security had to be questioned. However, as the

discipline was immature, and controlled by relatively few entities, questioning the status quo was not without difficulties.

However, in order for the discipline to mature, more feedback into the overall computer security (and in particular, the computer malware/virus) system was needed, requiring a shift from a mostly closed system to one that was forced to undergo greater scrutiny from various other communities. The input from these communities resulted in long-term changes and increased maturation of the system.

Figure 1 illustrates the specific areas in which the research presented herein addressed these needs, provides an overview of the research context, and outlines the specific impact of the research, specifically the development of new and significant scientific paradigms within the discipline.

The Generic Virus Writer	Application of psychology and sociology to virus writer motivation	Facilitated introduction of multidisciplinary approach to computer security dealing with malicious code to wider audience. Changed public perception of people involved in writing self-replicating programs. Provided course materials to further academic understanding of malicious code problem.
Virus Analysis: What a Winword Concept	Technical analysis of a new class of Malicious Mobile Code	Demonstrated viruses can spread via e-mail. Introduced the concept of upconversion.
Virus Analysis: Excel Yourself!	Technical analysis of the first Excel virus	Demonstrated viruses can spread via e-mail.
Virus Writers: End of the Innocence	Application of psychology and sociology to changes in virus writer profile	Laws have limited effect on virus writing. Demonstrated issues with artificial socio-technical divide.
The Antivirus Strategy System	Application of General System Theory to Virus Protection	Revisited idea that technical solutions are insufficient to solve the whole virus problem. Brought scientific discourse and radical ideas into the antivirus mainstream. Demonstrated how feedback within organization can impact overall system security.
What is Wild?	Review of virus testing methodologies and certification techniques	Documented insufficient standards and criteria. Defined valid and meaningful criteria and methodologies that were accepted by governments, testers, academic and public worldwide.
Cyberterrorism?	Multi-disciplinary approach to examining the role of Information technology in Terrorism	Vertical views of cyber-security are potentially dangerous, and reflect only one dimension of the threat. Creation of terrorism matrix
A Survey of Privacy Attitudes	Application of softer "human factors" in examining privacy problems	Quantitatively demonstrated that echnical solutions will not be successful without considering human factors.

Figure 1: Research, summary, and impact.

Introduction

For over a decade, I have been working in the field of computer science: specifically, those topics that involve the integration of other disciplines with the study of computer security. In this document, I propose that the body of work I have built up over the last eleven years is suitable for earning a PhD. based upon publication. In certain aspects of the discipline of computer security – in particular, within the field of computer viruses – the work is considered to be seminal [Lee & Harley, 2000; Rusch, 2002; Stucker, 1997].

During this time period, the work submitted and described herein has been published in peer–reviewed conference proceedings and journals, cited by many different academic and government entities, and included as part of university computer science curricula worldwide [Denning, 1999; Purdue, 2004; Kabay, 2001].

[Rogerson, 1996; Rogerson and Bynum, 1997] recognized the need to "build upon and elaborate the conceptual foundation" (of existing curricula dealing with computing ethics), at the same time "developing the frameworks within which practical action can occur, thus reducing the probability of unforeseen effects of information technology application". In a public letter addressing the content of a computer ethics course, [Bynum, 2003] called specifically for "Materials covering a representative sampling of "traditional" and "new" computer ethics topics like computer security (viruses, hacking, terrorism, etc.)".

The research presented herein provided such a framework in the field of computer security, and explored these 'new' computer ethics topics. Furthermore, it expanded the

existing technical worldview to encompass other areas critical to understanding computer security issues by approaching these areas with a multidisciplinary perspective. The acceptance of my research by the academic community as foundational, and its status as required reading in the area demonstrates its impact and as a scholarly work on the subject.

In summary, this research

- Critically examined the nature of the virus, hacking, antivirus and security communities at technical and social levels;
- Established, based upon the aforementioned examination, the need for a multidisciplinary approach in solving the computer security problem;
- Resulted in extensive discussion of human and technical issues of these communities in mainstream media, gaining worldwide attention and recognition;
- Built upon the worldwide attention and recognition via both private and public discussion, facilitating introduction of the concept of a multidisciplinary approach to areas of computer science dealing with malicious code;
- Facilitated the integration of a multidisciplinary approach to new computer security topics into the security community and into academic curricula.
- Facilitated the introduction of new paradigms in computer security resulting in the maturation of the discipline, i.e. the areas represented herein are now acceptable within the scope of study of computer security.

The structure of this context document is relatively straightforward. In the General Overview section, the impact and scholarly validity of my research is described. This section may be considered to be a justification of the application, and contains a discussion of the scope, quality and quantity of work. Next, the context of the work is considered – the existing state of the art at the time of carrying out much of the research is given, and it is placed in relation to contrasting approaches. This is followed by an overview of the actual research carried out.

Finally, further research areas identified by my work are highlighted; suggestions for different approaches are given. Additionally, some personal thoughts on the experience of conducting the research are shared, as well as the motivation for seeking this degree.

General Overview

When one considers the award of a PhD by publication, several important checks and balances are brought to mind to ensure that this mode of degree application is considered to be equivalent to a PhD obtained by a more traditional approach. The most important of these are comparability with traditional degrees, and literary record – that is, that the PhD be represented by some permanent record to which other scholars may point to in order to extend, compare and critique the underlying body of knowledge.

These preconditions can be further broken down by applying more traditional measures of the PhD: suitability, quality, quantity and scale/scope. In this overview section, it is my intention to demonstrate that the work I have published to date clearly satisfies each of these primary criteria.

The main thrust of all my work has been the integration of 'traditional' branches of science with the emerging field of computer security. As computing power has grown exponentially, the impact of computing technology has been huge. Radical changes in our spoken language ('dot.com', 'slashdotted', URL, etc.) are indicative of the tremendous revolution in which we have unwittingly taken part.

Due to the speed with which computing technology has developed, there has been little time within the computer security industry for issues aside from the raw technology itself to be considered [Gordon, 1994a; Yang, 2002, Yngstrom, 1996].

Much of the work presented here was begun at the very start of these changes: this understanding is *extremely* important in placing the work in its context.

In terms of comparability, discussion is somewhat lengthier; so I instead turn to the latter category of literary record. Here, I demonstrate that the volume of the publicly available work (made available through peer–reviewed publishing and other means), coupled with this context document is sufficient to provide a permanent, coherent record of this research.

The volume of work is distributed throughout several mediums: journal publication, white papers, conference presentations, classroom content and press (television, radio, and print media). Here I will discuss each of these areas in more detail. In the early days of computer viruses, while there were engineering and computer science journals, there

were few opportunities for publications related to computer viruses or security and almost no people working in the field of computer viruses and malware[1].

Elsevier Science's *Computers and Security Journal* was one of the earliest recognized specialist journals in the security field; it was fortunate to have as Senior Editors Dr. Harold Highland and Dr. Jon David. Highland and David were pioneers in computer antivirus research; thus, they were able to provide the scientific expertise to critically review submissions related to computer viruses.

It was in *Computers and Security Journal* that my first widely acclaimed and award winning work 'Technologically Enabled Crime: Shifting Paradigms for the Year 2000' was published. That work is discussed in detail later as one of the submissions for consideration. A second paper, 'Cyberterrorism?' also submitted herein, was later also published by the peer-reviewed journal.

Shortly after the inception of Personal Computer viruses in the wild[2], the British specialist journal *Virus Bulletin* emerged, and quickly became the only specialist virus publication recognized by computer anti-virus researchers; thus, publication in this journal was critical to gaining acceptance of the work within the relatively insular antivirus community. Additionally, it was the *only* forum in which research on computer

---

[1] The set of malicious code contains replicating code (viruses) and non-replicating code.

[2] "In the wild" viruses are those which have been found on real users computers, and that spread during the course of regular day-to-day operations.

viruses was regularly shared. Finally, the review board of *Virus Bulletin* consisted of world-recognized researchers in the field of computer viruses.

My research on virus writing and virus writers was initially published in *Virus Bulletin,* in an effort to dispel some of the myths concerning people who wrote self-replicating programs, and to foster relationships with others in the closed antivirus research field. Subsequently, I have published several technical articles in the journal, two of which were analyses of the first two known macro viruses and which are also submitted as part of this context statement.

Prior to publication of these virus analyses, the antivirus research community, and subsequently the world, did not believe that viruses could be spread through documents or spreadsheets. My discovery and analysis proved they could.

In addition to the published research, presentations based on the journal submissions and articles have been given in various venues, including:

- The American Association for the Advancement of Science (AAAS) Conference on Computer and Network Use and Abuse, Irvine, CA, 1993.
- Special Projects for The United Nations Headquarters, NY, NY, 1994, 1995
- Elsevier Science Compsec UK Conference, Westminster, UK, 1996, 1997, 2003
- The National Institute of Standards and Technology Conference, Baltimore, MD, 1996, 1997, 2000

- The Computer Security Institute Annual Conference, Chicago, IL, 1994, 1995, 1996. 2000 (Keynote), 2001.

- Working groups hosted by International Federation for Information Processing (IFIP). 1995, 1996, 2002

- Working groups hosted by SRI International, Crystal City, VA 2003

- Working groups hosted by FBI Behavioral Sciences Unit, Quantico, VA. 2004

In addition, my work has been cited and I have been profiled many times by diverse types of international news media as a scientific and credible source. An example of such a quote follows; a more complete list can be found in Appendix 3.

- *"...she is a first rate tecchie, with impeccable credentials...conducted research of a technical, educational and psychological natures.....she has a long association with computer security and technology in general..... "Remember the daughters of Daneaus – sentenced to forever draw water in a bucket of holes? To approach the problem from a solely technical angle is the modern day equivalent". (Irish Examiner, 2003).*

Thus, technical, social and psychological aspects of my research have been published in numerous academic journals, and are well cited; the research has been incorporated into diverse conference proceedings and served as the basis for many presentations to a variety of audiences; and, the press coverage of the research forms an extensive and permanent electronic and print record in archive sites. Considering these facts *in toto*, this work is clearly integrated into the permanent record that forms our societies 'body of knowledge'.

Comparability: Quality, Scale and Scope

The more difficult issue of comparability may be broken down into two areas: quality, and scale/scope. What follows shows that the work satisfies each of these categories.

Quality

In terms of quality, the work has been awarded prizes and been presented as 'invited work' at scholarly conferences. Some of these are:

- American Association for the Advancement of Science Conference on Social, Ethical and Legal Implications of Computer and Network Use and Abuse [AAAS, 1993];
- International Federation for Information Processing Sec 94 [IFIP, 1994];
- IFIP World Computer Congress [IFIP, 2002];
- European Institute for Computer Antivirus Research [EICAR, 1998]

As discussed previously the work has also been published in several journals, both peer and non–peer reviewed (e.g. *Computers and Security Journal; European Institute for Computer Antivirus Research, Network Security* and *Virus Bulletin*).

Additionally, attesting to the quality of the research is that it has been presented at academic and governmental research institutions, conferences and workshops worldwide, including:

- *University of Stockholm Department of Computer Science*

- *University of Hamburg Department of Computer Science*
- *University of Aalborg [hosted European Institute for Computer Antivirus Research]*
- *Purdue University Department of Computer Science*
- *Florida Institute of Technology Department of Computer Science, Center for Information Assurance*
- *University of Notre Dame Department of Computer Science*
- *Indiana University Department of Computer Science, sponsored by ACM.*
- *United States Department of Justice: FBI Behavioral Sciences Unit*
- *White House, Washington, D.C.*
- *United Nations, NY*

In addition to being of highest quality, my work on virus writers has become the seminal work in the field, and represents the foundation of all scientific virus writer studies currently conducted. Citing [Lee and Harley, 2002] from the *Best Paper Proceedings of the European Institute for Computer Antivirus Research Conference*:

> *"The online community with its relative anonymity and anarchic structure is an area in which personal ethics are to the fore, and the clash and mix of ideas could easily form the basis for many sociological and anthropological studies in the*

> *area. Notable for the seminal work in this area is Sarah Gordon, whose writing has provided the AV research community with many rich insights into the workings of the virus writer's mind.*"

In his book *Software Forensics: Collecting Evidence from the Scene of a Digital Crime,* [Slade, 2003] voices representative sentiments:

> "*Over the years, we have been able to glean ideas about the characteristics of this tribe. For this information, we are all indebted to researchers such as Sarah Gordon...*"

My work has been integrated into required course work for students in the Computer Science track studying Computer Security at several institutions, including *Florida Institute of Technology, Purdue University, Norwich University* and *Georgetown University.* Based upon the preceding evidence, I believe the question of quality is satisfied.

Scale and Scope

The issues of scale and scope are somewhat more nebulous – one instinctively knows suitable work when one sees it, but defining how much breadth is 'enough' can be difficult [Draper, 2002]. Here, I believe that the length of time the work has been published, coupled with the wide number of different disciplines covered provide evidence of sufficient scope to qualify for the degree of PhD.

Even though one could coherently argue that a PhD based upon just the work on virus writers would be acceptable, the body of work described covers technical computing assets of virus writers, hackers and terrorists, mainstream and journal articles on virus analysis, and the application of ethics, psychology, education and general systems theory to computer security.

Thus, in terms of scope, the work is far broader than a single paper but instead represents a manifesto for application of other disciplines to computer security, and integration of skill-sets to holistically address computer security issues.

Research Context

As outlined above, the evolving computer security industry has historically focused on the purely technical aspects of computers – moreover, within these technical aspects, application of holistic or multidisciplinary approaches while seen occasionally (Kephart and White, 1991], has been the exception [Gordon, 1995a].

Thus, at the time this research began, there was little or no attention given to computer security except at a binary–code level: the problem was primarily considered to be a problem of bits, bytes, and coding, no more, no less [Gordon, 1994a].

From a current perspective, the worldview of most computer security professionals and academics in the computer science field the early nineties seems narrow in the extreme: a mere decade and a half later it is now widely understood that the issues that are involved in providing for 'secure' computing are technical, legal, ethical, psychological and social: in essence, systemic and holistic.

Citing Gene Spafford, Director of Purdue University's Center for Education and Research in Information Assurance and Security, upon receipt of the 2000 National Computer Systems Security Award:

> *"...we need to rethink the research and education we perform in this area. We should be including psychology, management, economics, and sociology in what we do."*

This concept of security extending well beyond the technical and into other realms is an important aspect of the research presented herein. At the time the work began many security problems were seen as entirely technological in nature. However, even a cursory discussion of the problems of Trojan Horses sent by e-mail reveals that the role of the user is paramount: no amount of technology will ever stop the user from using his/her privileges to the determinant of security. The ultimate example of this was the JDBMGR hoax – an e-mail that instructed users to delete certain (actually useful) files from their machines; many users complied.

Clearly, psychology and education are vital pieces of the security puzzle. At the same time, psychology or technology alone will not solve security woes; the real solution requires working synergistically across multiple areas.

Technology

This need for a holistic approach reaches into all areas of computer security; here it will be clearly demonstrated in the evolution of the computer virus problem. While the formal definition of a computer virus is somewhat complex [Cohen, 1986] at the most

fundamental level, computer viruses are simply computer programs that have the property of self–replication.

While antivirus researchers continue to quibble over the *exact* definition of a virus [Highland, 1990; Websters, 2004], it is clear that self-replication is the only property that a computer program *must* have to fulfill the requirement for being a virus.

Self-replicating programs that attach to hosts are generally called *computer viruses*. Self-replicating programs that do not attach to hosts are generally referred to as *worms*. Self-replicating programs may or may not contain overtly damaging code; however, they often contain instructions that can indirectly and unintentionally damage data or introduce program or system/network instability. Computer viruses exist in the wild, or in the collections called zoos; computer users are at most risk from those viruses found in the wild.

Computer viruses fall clearly under the umbrella of 'technical' problems that suggest a technical solution. Thus, to approach the problem of computer viruses requires an understanding of how viruses work, how to defend against them at the technical level, and how to test that defense using scientifically sound methods and meaningful criteria.

Non-Technical Issues

Clearly, understanding technology is important for developing solutions; however, the problem is not solely technical: it is also a problem with social, psychological and ethical facets that introduce new questions

Social and Psychological Issues

A good example of the role of societal issues in computer viruses relates to how computer viruses are produced. Computer viruses do not appear by chance or accident; they are written by individuals, and sometimes, as a cooperative effort between one or more individuals. There are relatively small numbers of virus writers worldwide who choose to explicitly release computer viruses into the general computing population; however, many others experiment with viruses without explicitly releasing them [Gordon, 1994b].

The individuals often form groups. The formation and disintegration of these various virus-writing groups has been observed over time in the course of this research. Given that there are often just a handful of active groups and that a large percentage of the current 'virus problem' can be attributed to even just one person [Kotadia, 2004], gaining a better understanding of these individuals and groups could help us understand and dissuade virus writers and distributors.

At the social and psychological level, research related to groups involved in both the virus and antivirus worlds have been explored, and solutions for advancing both the state of computer security as well as the science of antivirus research put forth. The questions addressed and issues examined throughout my research[3] are diverse; some of the more notable are listed in the paragraphs below.

[3] Many of these are addressed in submitted publications; others are addressed in supplemental work referenced Appendix 2.

Who writes self-replicating programs? Why do they decide to release them? How do individuals become involved in virus writing groups? What might influence a participant in the virus writing subculture to stop writing viruses? These questions are explored and answered in "The Generic Virus Writer" and "The Generic Virus Writer II". The research dispelled the myths that virus writers were all unethical teens living in the basement waiting to destroy the world, and demonstrated that at least in some cases the young people who chose to write and make available their viruses a homogenous group, within ethical norms for their ages, and who drew a mental barrier between making a virus available[4] vs. releasing it.

How do these groups share information? Do the virus writers and hackers share information in the same way, or are there differences? Do groups tend to be national or international; localized or widely distributed? How have these groups influenced the production of computer viruses and their subsequent appearance in the computing population? These questions were addressed, and answered in *Technologically Enabled Crime: Shifting Paradigms for the Year 2000.* What, if any, benefit has come from the experiments of people who write computer viruses? This issue is addressed in *The Antivirus Strategy System.*

Ethical Issues

In addition to social and psychological issues, ethical issues must be considered when examining the problem of computer viruses.

---

[4] For example, a virus might be made available for public consumption on an FTP or WWW site.

Ethics and virus writers

For example, when considering the creation and availability of self–replicating code, what actions can be considered 'right', and what actions are 'wrong'? It is generally not illegal[5] to post self-replicating code in public forums provided that it is labeled as such; however, is it ethical? Is there any merit to the idea that such postings help secure systems by increasing public awareness? An open discussion of these issues amongst computer science professionals, and computer security professionals in particular, could go a long way toward arriving at answers which could be considered the standard or norm in terms of acceptability.

An informed discussion of the ethical issues can help other educators impart messages that act to dissuade irresponsibility in these areas, and that encourage responsibility and accountability. However, as computer security is a relatively young discipline, it suffers from many of the problems outlined in [Kuhn, 1970]. In particular, there has been a lack of agreement on which areas should be "allowed" to be discussed. The need for such work is large; I am personally aware of a Computer Science course where students were encouraged to develop viruses in insecure networks as part of their learning process [Gordon, 1996b]; however, this sort of activity is decreasing as awareness of the ethical issues increases.

---

[5] In the United States and in the United Kingdom.

Ethics and the antivirus community

Ethical issues are not just the purview of virus writers; antivirus researchers also face many difficult decisions that are in part related to ethics. For example, what is the most useful model for sharing self-replicating code? What rules should antivirus researchers follow when exchanging or sharing samples of computer viruses? What ethical issues are introduced by the insular nature of the antivirus research community, and does this nature help increase or damage the state of scientific and academic research in the field? What ethical issues arise in the process of replicating samples, or conducting and publishing research on new methods of infection or distribution? How are competitive issues related to not sharing samples balanced with the goal of providing protection for all? While these questions cannot yet be fully explored, the research presented in *When World Collide: Information Sharing in the Antivirus and Security Communities* offers the first exploration of the issues, and documents the wide divergence in attitudes between antivirus researchers, security professionals, and academics. This work, while not submitted as part of the PhD requirements, remains one of the standard pieces of work discussing differences in information sharing models between the antivirus and security practitioner.

Appropriate Legal Intervention

Children in primary and secondary schools now use the Internet; virtually 100 percent of schools are connected in the United States, Australia, Finland, Canada, and Great Britain with availability increasing in schools from Scandinavia to Israel and Korea [Schofield, 2003].

Given the potential for actions that can cause hundreds of millions of dollars damage from the child's own classroom, or in many cases, home, crossing federal and national boundaries in the process, many questions related to law emerge.

How can the virus problem be tackled from a legal perspective? What legal remedies are available that allow the punishment to fit the crimes? Is legal intervention even an effective deterrent for this type of activity? Both *The Generic Virus Writer* and *Virus Writers: The End of the Innocence* address these issues.

The legal aspect of the computer virus problem is made more complex by the multi–national nature of the problems faced. A particular virus may be written in a jurisdiction where it is *legal* to write, possess and make available clearly labeled computer viruses with subsequent damage occurring in a place where the writing, or making available of computer viruses is illegal.

Clearly a significant part of the virus problem falls within the realm of actions that are legal, and extremely complex; however, for many years it was assumed by many in the computer virus field that laws written to 'simply lock them all up' or 'make virus writing illegal' could solve the problem. This sentiment became dangerously close to reality following statements by [Tippett, 2000]. Tippett, Chief Technologist of the *International Computer Security Associations,* urged Congress to make virus-writing itself a crime, stating "I would suggest that we make this one of those few First Amendment exceptions and make it illegal to create them."

Such a statement is a dangerous oversimplification; it is outrageous to think that any and all self-replicating code should be outlawed. For example, given that the definition of a

virus centers on the property of self-replication, an installer than copies a copy of itself to new media would technically be a virus and therefore outlawed under the First Amendment. Furthermore, there are often cases where a virus research must legitimately create a new virus.

For example, periodically, virus writers create 'virus construction kits' – automatic systems that allow non-technical users to create 'new' viruses. While reverse engineering and studying these kits is one approach, a more rapid and perhaps more pragmatic approach is to use the kits to create a large number of self-replicating programs and study them for similarities. Once this is done, these 'new' viruses would then be deleted. Should such a pragmatic solution be outlawed under the First Amendment?

Finally, the reduction of the area protected by free speech is a last-ditch approach to solving the problem: it is the most serious censure that can be given. Existing laws must be carefully examined; for example, those concerning carelessness or reckless endangerment may offer other approaches that could be used in addressing computer crime.

Other non-technical aspects

Finally, even at a preventative level, an effective solution is likely to have significant non–technical components. [Gordon, 1997] describes the Christma.EXEC worm of 1987. This worm required users to cut and paste the code from an email and execute it – thus the worm relied on significant user action in order to spread. Similarly, the recent outbreak of the Novarg worm (also known as W32/MyDoom) relied on users double–

clicking on attachments, thereby infecting their machines and continuing the spread of the worm.

Some viruses have even gone a step further, compressing infected files in password-protected archives, and sending the password to the user via email. Naïve users meekly follow the on-screen instructions, enter the password and thereby infect their own machine. In every case, the virus relies on non-technical factors to facilitate spread.

Virus writers themselves are not ignorant of the power of social engineering[6] [Gordon, 1995b]. Mass mailing worms and viruses have used a number of different approaches to increase their probability of propagation, leveraging such strong concepts as sex and danger. These motifs are highly memetic, and led us to consider another important aspect of my work.

In [Gordon, Ford & Wells, 1997], Dawkins' meme theory [Dawkins, 1989] is expanded upon, and the creation of successful hoaxes is examined in the context of leveraging powerful memetic themes. This work, which integrated technology with psychology, was useful in that it indicated that widespread distribution of a 'vaccine' could significantly suppress the memetic nature of such hoaxes.

---

[6] Social engineering is intentionally manipulating a persons responses to achieve a goal; it relies on people's desire to be helpful or be trustful of or obedient to authority.

Expanding Worldviews

These issues cannot be reduced to purely technological problems; rather, they touch on psychology, user intent and the human–computer interface. Therefore, to fully understand the problem and offer an integrated, comprehensive solution, a multi-disciplinary approach must be taken.

When this work was started, little prior research had been carried out into these non–technical but related issues. Thus, in many ways there was little precedent for the areas that I explored. This exploration was not without challenges, related both to gender as well as a culture that did not wish to discuss, let alone integrate, new ideas – especially ideas that questioned the very premises of the culture [Gordon & Ford, 1999]. A study of contrasts between the security and antivirus worlds illustrates various outcomes of this Descartian worldview in some detail [Gordon & Ford, 1999] [7].

Diagram 1 provides a graphical illustration of antivirus research as a function of computer security prior to this research. Diagram 2 provides a graphical illustration of this research's impact upon malicious code research as a function of security as submitted in this documentation.

---

[7] *When World Collide* describes the insular nature of the community and the resultant polarization from other scientists, and shows how this isolation has stifled true scientific exploration of many of the most pressing issues.

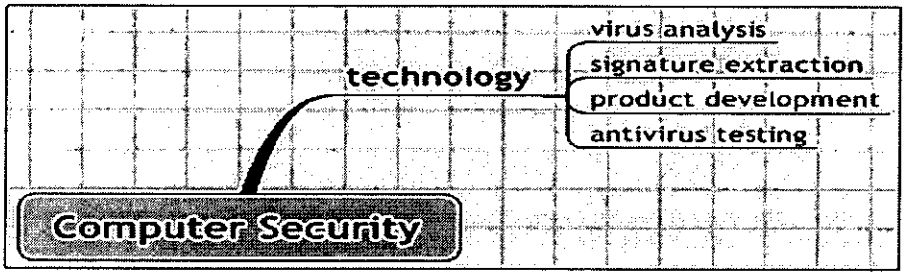


Diagram 1: State of perceptions of the malicious code arm of computer security prior to my research impact.

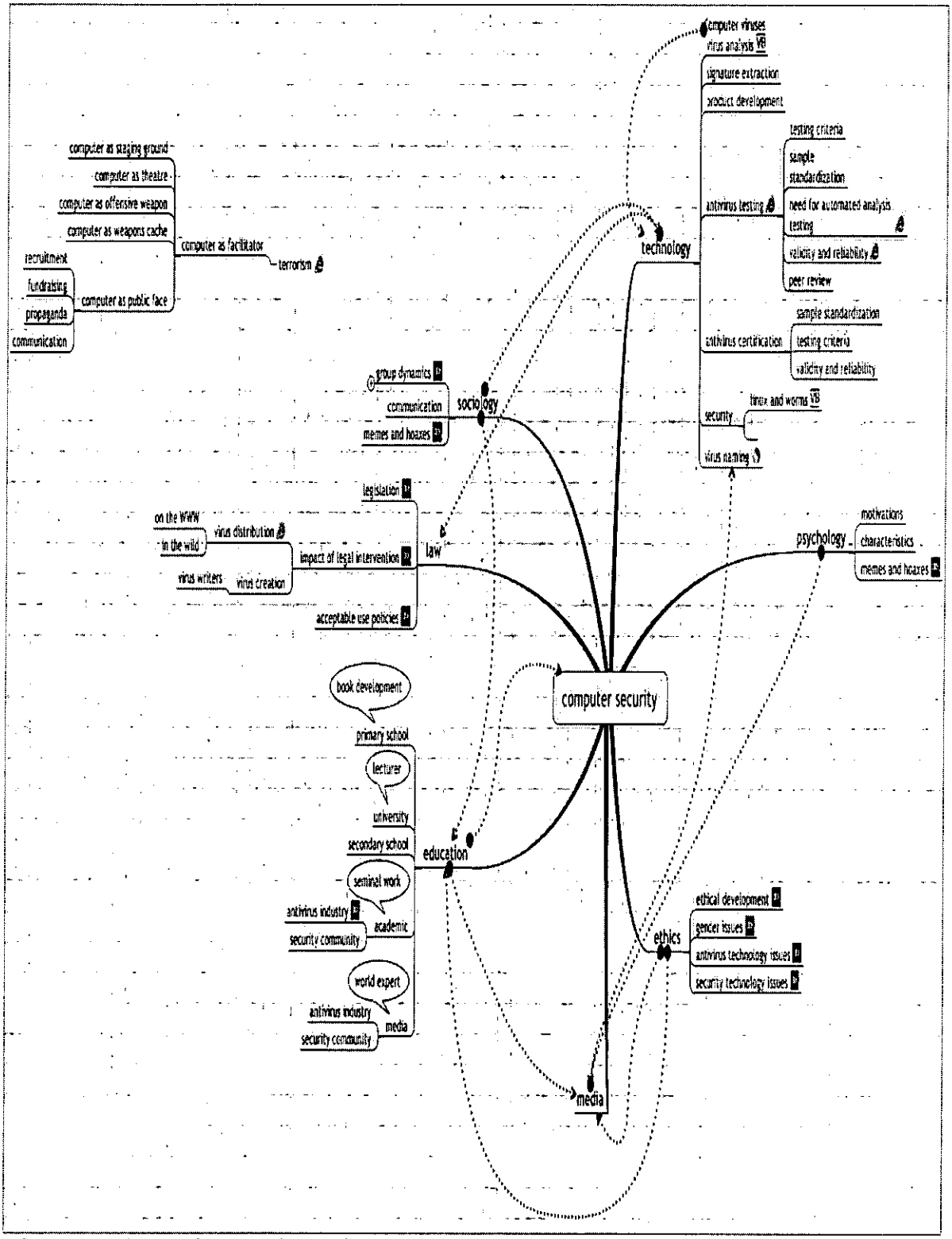


Diagram 2: State of perceptions of computer security after impact of my contributions.

As illustrated by the partial overview of the research in Diagram 2, the traditional reductionist approach to solving computer security problems – dissecting them into their smallest parts and studying those parts in relative isolation – is receding into history, replaced with a worldview that is neither humanistic nor mechanistic, but a realistic, useful and ultimately appealing mix of the two. Once the sole domain of computer engineers, computer security is now being examined not only as a technological issue, but as an issue that must be addressed by law, ethics, education, psychology and sociology as well.

An Important Caveat

At the same time, attempting this integration requires great care to avoid drawing inappropriate conclusions. A good example of confusion caused by the integration of multidisciplinary techniques comes from the area of computer viruses. Here we often hear of the problem of computer monoculture – that is, that the global system is vulnerable because of the high degree of homogeneity between systems.

Often, an analogue to biological systems is cited, where a monoculture can easily lead to species extinction [Geer, et. al., 2003]. However, there are important differences between virus spread in a biological system and a computer network. Perhaps most importantly, the 'extinction' threshold between a living system and a computer system is very different.

While in a biological system a very large percentage of a species needs to be destroyed to threaten species survival, the Internet can be damaged beyond use by the impediment or death of just a handful of systems – for example, the root DNS servers [Ford, 2004b].

Thus, in this case, application of a concept borrowed from another field can lead to inappropriate or misleading conclusions.

While there is risk involved in such borrowing, my work has shown is also significant benefit. The most important caveat is that one must appropriately consider the applicability of results obtained from a multi-disciplinary approach.

Can the results be verified experimentally? Can the link between the two disciplines be verified using some quantitative means? Are there any underlying assumptions that are not correct in the system of study? What can be done to improve the fit between the two different approaches used? By carefully considering each of the preceding questions it is possible to improve the results obtained by multidisciplinary work.

Furthermore, it is important to consider the limit of applicability of different techniques. For example, one cannot create a virus signature with a psychology test, and one cannot improve data security by simply studying privacy through the years. Multidisciplinary techniques have very real limits and these must be considered when applying techniques to the security problem.

Research Submission Overview

The following papers are submitted as part of this Context Statement.

Paper 1: Gordon, S. 1994a. *Technologically Enabled Crime: Shifting Paradigms for the Year 2000.* Computers and Security Journal. October. 1994. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/Crime.html

Paper 2: Gordon, S. 1994b. *The Generic Virus Writer.* From the Proceedings of the Fourth International Virus Bulletin Conference. Jersey, U.K. September 1994. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/GenericVirusWriter.html

Paper 3: Gordon, S. 1995a. *Virus Analysis: What a Winword Concept.* Virus Bulletin. September 1995. Also available from: http://www.virusbtn.com/magazine/archives/pdf/1995/199509.PDF

Paper 4: Gordon, S. 1996. Virus Analysis: *Excel Yourself!* Virus Bulletin. Also available from: http://www.virusbtn.com/magazine/archives/pdf/1996/199608.PDF

Paper 5: Gordon, S. 2000. *Virus Writers: End of the Innocence.* From the Proceedings of the International Virus Bulletin Conference. Orlando, Florida. Also available from: http://www.research.ibm.com/antivirus/SciPapers/VB2000SG.htm

Paper 6: Gordon, S. 1995d. *The Antivirus Strategy System.* From the Proceedings of the International Virus Bulletin Conference, Boston, MA. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/Strategy.html

Paper 7: Gordon, 1997. *What is Wild?* From the Proceedings of the 1997 National Systems Security Conference. National Institute of Standards and Technology. Baltimore, MD. . Available from http://csrc.nist.gov/nissc/1997/proceedings/177.pdf

Paper 8: Gordon, S. & Ford, R. 2002b. *Cyberterrorism?* Computers & Security 21(7): pp. 636–647 2002. Also available from: http://www.compseconline.com/premium_article/premcs.htm#vol21issue72, and from http://securityresponse.symantec.com/avcenter/reference/cyberterrorism.pdf

Paper 9: Gordon, S. 2003. *A Survey of Privacy Attitudes and Operational Behaviours in US, UK and EU Information Security Professionals.* Keynote Presentation. From the Proceedings of the Compsec 2003 Conference. Queen Elizabeth II Center. London, United Kingdom. Also available from: http://securityresponse.symantec.com/avcenter/reference/privacy.attitudes.behaviors.pdf

Predicting our Future: Technologically Enabled Crime

Paper 1: Gordon, S. 1994a. *Technologically Enabled Crime: Shifting Paradigms for the Year 2000.* Computers and Security Journal. October. 1994. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/Crime.html

The first paper that I would like to put forth for consideration is 'Technologically Enabled Crime: Shifting Paradigms for the Year 2000' [Gordon, 1994a][8], presented at the IFIP Sec 94 Conference, and published in *Computers and Security Journal*. The paper explored the need for the integration of ethics into various aspects of technology by examining social and ethical factors involved in the transmission of computer viruses and other malicious software, as well as the systems and technology.

[8] Available online at http://www.research.ibm.com/antivirus/SciPapers/Gordon/Crime.html

By drawing parallels between the development of the Internet and security technology with the development of medical technology, the work explores the potential for harm in doing things 'because we can', and considers similarities between the developing Internet and the need for focus on ethics with the integration of ethics with other scientific disciplines.

Technologically Enabled Crime explored ways in which the nature of the technology contributes to its own misuse, instability and potentially destruction, and considered the implications of such development. The summary concluded: "Without the proper interaction of laws, education and ethical development, there is a very real risk that this technology will soon become unusable and ultimately self–destructive". The paper's grim prophecy has been almost realized in the last year, when network–aware malware has shown its ability to cause worldwide network instability in surprisingly short time frames [Moore, 2003].

This paper was highly controversial at the time of publication – something that seems surprising given the current focus on interdisciplinary research within the security space. However, at the time it was written, the paper was radical in that it introduced 'soft' issues to the otherwise binary world of virus prevention. Current trends in virus research were focusing on increasingly technical solutions; this paper was really the first of its kind to view the problem holistically.

By focusing on the ethical, legal, and social aspects of these virus-related activities, a broader understanding of the real factors that affect computer security and specifically virus spread could be considered for the first time.

There was heated debate at the conference as to whether or not there was a need for computer science students to be exposed to the idea of ethics; several prominent computer scientists felt this was solely a technical issue [Highland, 1995].

However, the holistic approach to security introduced by the paper prevailed, and this work won the conference 'Best Paper' award. This view of security is now widely established at the correct way of considering the security of a system as a whole.

Virus Writers – Unraveling the Mystery

Paper 2: Gordon, S. 1994b. *The Generic Virus Writer.* From the Proceedings of the Fourth International Virus Bulletin Conference. Jersey, U.K. September 1994. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/GenericVirusWriter.html

This work [Gordon, 1994b] 'The Generic Virus Writer'[9], was first presented in September 1994 at *The Virus Bulletin Conference* in Jersey, United Kingdom. This paper debunked the myths surrounding virus writers, exploring their real motivations, as well as their ethical development.

At the time of the research, very little was known about the individuals and groups that were responsible for virus writing. Speculation was that they were young, and a fairly homogenous group; unethical, and socially challenged. However, these opinions were little more than prejudices: in reality, no scientific work related to virus writers had been

[9] http://www.research.ibm.com/antivirus/SciPapers/Gordon/GenericVirusWriter.html

done, and the statements that were made were generally made *ad hominen* rather than by focusing on any scientific or academic analysis of the individuals involved. In order to shed some light on this world, a methodologically sound course of investigation was employed, to search for the 'generic' virus writer. The research goal was to determine if the population of active virus writers was indeed a homogenous group as purported by the people involved in protecting systems from virus infection. In order to support the hypothesis that *all* virus writers who were engaged in this activity were unethical and malicious, it was necessary that no exceptions to the case be found. The inductive analytical method was employed for the actual analyses; the instrument used was based on recognized methods for assessing cognitive and ethical development.

I was uniquely qualified to carry out this research. During the preceding years, I had become well known in the computer underground as honest and trustworthy. This building of this trust began when I offered a communication network free for public use: VFR Systems Bulletin Board (BBS).

The BBS operated initially over the FIDONET network. The FIDONET allowed computer users worldwide to connect to BBS globally, connected by dial-up modems. The users could chat with each other and/or the system operator (SYSOP), leave messages in public forums called 'FIDONET Message Bases', or send personal e-mail using 'FIDONET Mail'. The Message Bases were focused on various topics including bases germane to hacking and virus writing; specifically, bases concerned with assembly language programming, UNIX, and computer viruses. Eventually VFR Systems expanded to use the Internet.

Within the confines of the BBS, many discussions took place amongst people who were active in the early Internet underground subculture. The discussions were generally private, technical in nature and established credibility with this growing technically focused population.

Participation at hacker conferences such as DEFCON (as a regular speaker and panelist) and HOPE[10] (as a delegate) and attendance at meetings of hackers at 2600[11] meetings worldwide, the trust which began in the early 1980s continued to build throughout the ensuing years. I became recognized as a technically competent person who, while I may not agree with individuals' choices of computing behaviors, did not personally judge the individual (rather, only their behaviour). I was outspoken regarding my thoughts on ethical issues related to computer viruses and hacking, and was asked to contribute dialogue with an 'alternative point of view' to some of the underground journals and projects. A potent example can be found in [HEX, 1992]. However, rather than my strong stance on the problems caused by viruses or hacking creating problems related to my trust, my honesty reinforced the fact that I had no hidden agenda. I was simply interested in the topic at hand.

While other members of the antivirus industry engaged in volatile and non-productive dialogue with hackers and virus writers, world-renowned journalist James Hattori had this to say about my work:

---

[10] HOPE is "Hackers On Planet Earth", a conference for hackers held annually in New York.

[11] 2600 is a well-organized group that promotes monthly meeting of hackers worldwide..

> *"Gordon is an anomaly in the anti-virus industry.... Sarah has spent years interviewing, profiling and exchanging ideas with virus writers, and in turn, they talk to her and trust her"* [CNN, 2000]

Although I was part of the security and anti–virus community, many of the virus writers of the time were known to me electronically, and I was well versed in the virus writer culture. Thus, during the early days of computer viruses when there were less than a hundred computer viruses circulating in the wild, I was able to interact with many people involved in the virus writing community. Of these interactions, only 3 were negative: i.e., refused to answer and responded with hostility.[12]

The study began with elicitation for participation amongst the available population. The request for participation was distributed electronically via the FIDONET Virus and Virus–Info Echo-mail[13] system, and by word–of–mouth.

Potential respondents were provided with assurance of anonymity, and the opportunity to provide responses via e–mail or paper mail. Four of the respondents chose to respond anonymously via paper mail; those in-depth responses are archived along with the electronic responses. I also spoke subsequently in person and on the telephone with three of these four respondents. Other respondents chose to answer electronically; these

---

[12] The hostility consisted of threatening bodily harm: threats to break my kneecaps, and to set off a bomb in my mouth.

[13] Echo-mail was a mail relay system that allowed messages to be sent from FIDONET system to FIDONET system, usually via modem connections.

responses were further investigated to ensure the respondents were active participants in the community.

The methodology of the survey was straightforward. Various questions were given to assess the individuals' level of ethical development and reasoning ability. Subjects were ranked on the Kohlberg scale [Kohlberg, 1984] in order to determine their level of ethical development; other questions explored the age of the subject, and their peer and superior relationships. The population of virus writers in the community during this time was quite small; less than 100 active virus writers; in addition to the survey, four individuals were chosen for additional in–depth follow–up interviews.[14]

The results of the research were unexpected, and led to some surprising conclusions. While virus writers had been considered unethical delinquents, this research showed that many were in fact within ethical norms for their ages, and had normal relationships with their peers and parents. It is considered seminal work and is now required reading in many university computer security programs [Denning, 1999; Kabay, 2001; Purdue, 2004, Ford, 2004a].

---

[14] These four individuals were also part of [Gordon, 1996] which bore out the predictions of this study – the three young people aged out of the behaviours; the ethically abnormal adult continued to take part in virus related activities.

The First Macro Viruses: Concept and Excel

Paper 3: Gordon, S. 1995a. *Virus Analysis: What a WinWord Concept.* Virus Bulletin. September 1995. Also available from: http://www.virusbtn.com/magazine/archives/pdf/1995/199509.PDF

Paper 4: Gordon, S. 1996. Virus Analysis: *Excel Yourself!* Virus Bulletin. Also available from: http://www.virusbtn.com/magazine/archives/pdf/1996/199608.PDF

Gaining access to most of the world's virus writing population was a process that took approximately five years, during which time technically competent as well as trustworthiness was established. This involved participating in technical discussion groups as outlined above, and publishing a number of technical security papers including [Gordon, 1995a; Gordon, 1995b; Gordon, 1996; Gordon, 1998a; Gordon, 1998b]. 1 submit two of those analyses for consideration, in order to demonstrate technical excellence in the area of viruses. In addition, these two analysis represent a new chapter in the world of viruses. While viruses were thought to not be e-mail replicative, this worked proved that indeed viruses can and did spread via e-mail. It also introduced the issue of upconversion, which proved to be quite controversial. This is explored in the analysis below.

A New Concept

The first paper, [Gordon, 1995a] 'What a (WinWord) Concept'[15], published in *Virus Bulletin* is an analysis of the first Microsoft Word macro virus found in the wild. Along

[15] http://www.virusbtn.com/magazine/archives/pdf/1995/199509.PDF

with an analysis of the first Excel macro virus, [Gordon, 1996] 'Excel Yourself'[16], this work constitutes computer security's first and foundational work in analysis of macro viruses. These viruses were replicated, cures created, and the viral replicants were shared with antivirus product developers; the original samples are stored in a secure virus lab. This work is particularly noteworthy as it demonstrated for the first time that viruses could, despite prevailing opinion, be spread via e-mail.

New Ethical Questions

In addition to the technical issues, the virus analyses led to the exploration of some new and challenging ethical issues. As described above, in the case of the Concept virus, whilst it was initially reported performing replication within one file format (Office 95), my analysis showed it was capable of replicating within another file format (Office 98). This process was named 'upconversion'.

The process of upconversion is interesting, and gaining an understanding of the technical aspects of the process is helpful in understanding the ethical issues associated with the process. Early versions of Office used a different macro language from later versions – thus, macros written for early versions of Microsoft Word would not naturally execute under current versions of the program. In order to preserve backward compatibility, Microsoft introduced the process of upconversion, whereby newer versions of office would automatically 'rewrite' old style macros into the new macro language.

---

[16] http://www.virusbtn.com/magazine/archives/pdf/1996/199608.PDF

This practice was very helpful for many users, as it significantly eased the migration costs from obsolete versions of office. However, it was not just benign macros that could be rewritten; old macro viruses were also upconverted. Furthermore, signature based virus detection was unable to detect such upconverted viruses.

The question became whether such upconverted viruses were 'new' viruses, and if the process of deliberately creating upconverted viruses for the process of ensuring detection and remediation was 'unethical' as it constituted 'virus writing'.

At that time (1995–8) some technologists working within the antivirus industry believed the act of upconversion simply resulted in a representation of the original virus in a different way, and that there was nothing unethical about the technical process of upconversion. Indeed, it could be argued that such practice was important for the protection of vulnerable systems, and it would be unethical *not* to carry out the upconversion process.

For example, if a macro virus was found in the wild attached to a Word 6 file, but the analysis showed it could spread to and replicate on Word 7 files, developers of antivirus software would be remiss not to include Word 7 file protection.

[Chess, 2002] concurs: "*Upconversions of Word 6/Word 7 macro viruses clearly constitute a threat to the public, and anti-virus workers have a responsibility to address that threat.*"

Yet other technologists believed such additional replication on platforms other than the 'original' tantamount to 'virus creation', claiming this process resulted in 'new' viruses.

Creation of 'new viruses' under *any* circumstances was unethical, according to the beliefs of some researchers. Because of these beliefs, anyone who advocated upconversion was deemed by these technologists to be 'creating new viruses' and therefore 'unethical'.

In some cases, people who advocated or performed upconversion were verbally harassed, maligned, slandered, libeled, and their employers contacted regarding 'the unethical conduct of the upconverter'; journalists were contacted to 'expose' the 'unethical upconverter' [Gordon, 1998c; Bridwell, 2004; Saarinen, 1998]. In one case, a researcher was threatened publicly with physical assault for publicly supporting the concept of upconversion[Anonymous, 2004].[17]

The primary ethical debate remained more of a monologue – prescriptive in nature, with a few individuals appearing to vociferously dictate acceptable practice for the rest of the community. Attempts made by colleagues to engage in a reasoned debate about the ethical issues were ignored. However, within the antivirus community, these seemingly negative events led to the development of an undercurrent of resentment toward the predominating prescriptive worldview. Today, replicating virus samples on any available platform is routine, and seen by most responsible scientists as an ethical course of action.

However, the foundational dialogue on the acceptability of such practices was clearly laid with my willingness to counter prescriptivism and dogma on these issues, and with the

[17] Many witnesses to these occurrences state they are hesitant to come forth due to fear of retaliation. However, they gave consent to be listed as "Anonymous".

discovery and analysis of these early macro viruses, Concept and Excel, as presented in the two papers submitted with this Context Statement.

End of the Innocence: Changes in the Virus Writing Population?

Paper 5: Gordon, S. 2000. *Virus writers: End of the Innocence.* From the Proceedings of the International Virus Bulletin Conference. Orlando, Florida. Also available from: http://www.research.ibm.com/antivirus/SciPapers/VB2000SG.htm

When viruses are released, accidentally or purposely, into the general computing, the virus is considered to be 'in the wild'', i.e. it can be found spreading amongst the computer or network of computer users in the course of normal day to day operations. Other times, however, viruses are made available on the via vX [18] WWW or FTP sites. [Gordon, 1993] supported the hypothesis that viruses found on vX Bulletin Board Systems were not likely to end up in the wild; however, they remain items of interest for antivirus companies and end users alike, who pay the costs associated with their mere availability.

In addition to technical solutions for viruses, legal intervention had sometimes been suggested as a remedy for the virus problem. Thus, the fifth paper I will submit, [Gordon, 2000] 'Virus Writers: The End of The Innocence'[19], explores the perceptions of both the security community and the virus writers on the potential deterrent effect of legal

---

[18] vX is a representation of "Virus Exchange"; the term was created by this author and is in common usage in antivirus research..

[19] http://www.research.ibm.com/antivirus/SciPapers/VB2000SG.htm

intervention on virus writing. Additionally, the research analyzed the impact of various types of legal intervention on virus availability both on the WWW and in the wild.

This research examined the impact of high–profile legal intervention such as police raids, arrests and sentencing of virus writers and distributors on the number of viruses found in the wild, and on the WWW. I believed initially that legal intervention had not worked for several reasons, from the length of time between arrest and sentencing, to the age of the virus writer. This is consistent with social learning theory [Bandura, 1969] and ethical development [Kohlberg, 1984]. Simply put, by the time arrests or consequences had occurred within the community, the original members of the community had aged out and left the community; the new community members had little or no connection with the object of arrest/sentencing.

To explore any potential differences in the worldview of the people proposing legal intervention (antivirus researchers) and those at whom the interventions were aimed (virus writers and virus distributors), surveys of several populations were conducted: known virus writers, anti–virus researchers, and the population at the well–known DEFCON Conference in Las Vegas which included both virus writers and hackers.

The surveys were distributed to the antivirus researchers, virus writers and virus distributor groups via e-mail, and to DEFCON attendees in-person. In-person interviews were conducted early in the day at DEFCON, to overcome the hurdle of participants being under the influence of alcohol or drugs during the administration of the survey. The data is archived at IBM Research.

The findings indicated that many people who would not ordinarily consider writing a self replicating program felt that if it became illegal to do so, they would be *more* likely to do it, based on a desire to protect their right to do as they wish on their own computers, in their own homes. This was a particularly interesting finding as the 'right to free speech' was cited anecdotally by participants from many countries – some of which do not have a constitutional right to free speech.

The results were interesting and showed a marked difference between the communities. When asked how much laws and sentencing would alter behavior, the virus writers unanimously stated it would not, whereas anti–virus researchers were evenly split between yes and no. The evidence reinforced the virus writer position, as no discontinuity in the rate of discovery of new viruses in the wild is correlated with the arrest and prosecution of high–profile cases. If anything, the rate has continued to increase since the WildList began. [20]

Similarly, the DEFCON data showed some interesting trends. Here, hackers were randomly sampled during day 1 of the conference. There was expressed a very mixed view of the effectiveness of new laws. Many diverse comments were received, spanning the gamut from "yes – laws will be effective" to expressions of support for virus writing should it ever become illegal in and of itself.

---

[20] See **Research Overview** *Toward More Meaningful Tests of Antivirus Software* for more in-depth description of The WildList.

In addition to being released in Conference Proceedings, this presentation was filmed by CNN International and released as part of a special profile of my life and my research in this area. It is now a permanent part of the CNN International archive and is documented in Appendix 3.

The conclusion of the work was that there is little evidence of a deterrent effect of high–profile legislation or legal interventions. Overall, the research on virus writers resulted in an overturning of stereotypes, which in turn led to a deeper understanding of the computer virus problem.

Numerous interviews with virus writers reaffirmed that many of them do not see the impact of their virtual action on people in the real world. Lack of contextual clues, combined with depersonalization and desensitization that occurs in the online setting results in young people who appear pretty-much normal in all other ways, and who would not create the havoc a virus can cause to people they might encounter physically.

At the same time, young people were receiving much praise from the media for writing computer viruses, with their work presented as artificial life [Ludwig, 1997], art or digital graffiti [Dibbell, 1996]. In some cases, in addition to the praise from the media, virus writers received positive reinforcement such as employment [Middleton, 2001].

Overturning the stereotypes and doing research in previously uncharted waters afforded me the opportunity to speak with the media, and reinforce the fact that writing a virus is not 'rocket science', nor is it 'cool'; rather, it is unscientific and unethical to work with self–replicating programs in uncontrolled environments.

The impact of this study was broad. First, this work resulted in opportunities to educate the public, serving as a catalyst for discussion of clarification of societal views on acceptable behaviors in cyberspace (Appendix 3). It has led to a shift in questions asked by the media, and the media approach toward viruses in general. Now, there are few, if any, characterizations of viruses as 'digital graffiti', or 'artificial intelligence'; rather there are now discussions about responsibility. WWW sites that once hosted virus distribution now have acceptable use policies forbidding the distribution of viruses.

Rather than accepting student proposals for research based on creation of self-replicating code in insecure environments, university professors are now creating secure research laboratories, where students take responsibility for their work [Aycock, 2004].

While work related to virus writers was the focus of most media attention given my research, the integration of other disciplines into the way we develop, use, and view technology was not limited to virus writers only. The next papers discuss other applications of the approach.

A Holistic Strategy for Virus Prevention

Paper 6: Gordon, S. 1995d. *The Antivirus Strategy System*. From the Proceedings of the International Virus Bulletin Conference, Boston, MA. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/Strategy.html

The sixth paper I would like to submit is [Gordon, 1995d] 'The Antivirus Strategy System'[21]. With this work, a general systems theory model was applied to the world of

[21] http://www.research.ibm.com/antivirus/SciPapers/Gordon/Strategy.html

computer viruses. The goal of this application was to move the community's focus from a narrow view of computer virus prevention – which is highly focused on detection and disinfection of infected objects - to examining virus spread as a more general function of the overall system.

Approaches to the virus problem have been historically not only solely technical, but primarily reactive in nature [White, 1998]. This research offered a new way of looking at the interdependence between computer systems and security professionals, by examining causal factors rather than the traditional symptomatic relief.

Although the paper contained many concrete suggestions for improving virus protection the goal of the paper was to provide readers with a way to critically reassess their own systems, and not rely solely on technology to solve the problem.

Taking this broader approach allows administrators to include other issues in their anti-virus 'system' that were often ignored by those focusing just on core technology issues. As the paper argued, a holistic approach *must* be taken to virus protection in order to be successful.

Toward More Meaningful Tests of Antivirus Software

Paper 7: Gordon, 1997. *What is Wild?* From the Proceedings of the 1997 National Systems Security Conference. National Institute of Standards and Technology. Baltimore, MD. Available from http://csrc.nist.gov/nissc/1997/proceedings/177.pdf

The seventh paper I submit, [Gordon 1997] What is Wild[22], continued earlier work to change the worldview of the computer security world to include scientific testing criteria and methodology for antivirus software tests.

Heretofore, testing of antivirus software was done with little, if any, documentation, and somewhat arbitrary criteria. Detection of all samples was treated equally, even though the chances of infection by certain zoo viruses was almost zero.

This paper proposed a new, scientific approach to the testing of antivirus software work, and was conducted on behalf of IBM's Thomas J. Watson Research Center and was presented at the National Information Systems Security Conference.

By querying each antivirus testing body and exploring the ways testing was currently being done and exposing shortcomings, critical and necessary groundwork for future work in antivirus software testing was done. By establishing a precedent for scientifically valid, reproducible tests, the foundation for meaningful tests using real, well-maintained collections of viruses was created.

This work was furthered by establishment of The WildList Organization International[23], as well as by solicited work from the National Computer Security Center on antivirus software testing [Gordon & Ford, 1996; Gordon & Howard, 2000].

---

[22] http://csrc.nist.gov/nissc/1997/proceedings/177.pdf

[23] http://www.wildlist.org/faq.htm

Initially while there were only a few viruses found in the wild, antivirus product technology was aimed at both wild and zoo viruses; the technology was marketed according. Tests that were performed assessed product ability to detect the huge zoo collections as well as the in the wild viruses.

There was no method by which viruses actually spreading could be assessed. In 1993, Joseph Wells began to keep track of viruses actually seen by researchers in the wild, and formalized his reports in a document called *The WildList*. However, the reports represented only a small number of researchers and product developers.

In 1995, during collaboration with Wells, the idea of forming a larger, but still independent, formalized organization to monitor the spread of viruses emerged. He agreed, and the resultant efforts led to the formation of The WildList Organization, International in 1996.

In addition to tracking virus outbreaks, we wished to provide a way for researchers and testers to know that the viruses they had actually *were* the same viruses that were spreading in the wild. Thus, we created *WildCore,* the first formal set of sample replicants generated from viruses provided to *The WildList Organization, International* reporters. *WildCore* provided testers for the first time a reference set of those viruses that were actually likely to be found in the 'real' world.

The *WildCore* sample set has become the baseline criteria for *all* antivirus product certification efforts, including those developed by The University of Hamburg, The University of Magdeburg, The University of Tampere, Florida Institute of Technology, The International Computer Security Association, Virus Bulletin Certification, West

Coast Labs Secure Computing Checkmark, and UK ITSEC Certification. Generation of the sample sets was later transferred to antivirus and security expert Ian Whalley, IBM Research and then to antivirus researcher Shane Coursen.

The *WildList Organization International* operated as a non-vendor-affiliated collaborative effort of all major antivirus product developers, testers and independent researchers, until 2001 when ownership of the *WildList Organization* was transferred to Tru-Secure Corporation.

WildCore continues to be replicated and distributed to vendors and testers. The set is currently the baseline detection requirement for all antivirus technologies, and is used by testers worldwide. It is now managed by The International Computer Security Association (ICSA), a for-profit corporation in Carlisle Pennsylvania.

Additionally, the testing research undertaken resulted in my significant contributs to the development of the UK ITSEC model for antivirus software via the CESG[24] antivirus working group, and on conjoint development of testing criteria for the National Computer Security Association with Dr. Richard Ford[25]. Most recently, the outgrowth of this work [Gordon, 2002a] culminated in a joint project with a US Government organization dealing with virus analysis and classification. [26]

---

[24] CESG is the UK National Technical Authority for Information Assurance.

[25] http://www.malware.org/resume.htm

[26] This project is under development; details will be provided in subsequent publications.

Cyberterrorism: Fact or Fiction?

Paper 8: Gordon, S. & Ford, R. 2002b. *Cyberterrorism?* Computers & Security 21(7): pp. 636–647 2002. Also available from: http://www.compseconline.com/premium_article/premcs.htm#vol21issue72, and from http://securityresponse.symantec.com/avcenter/reference/cyberterrorism.pdf

With research spanning 2000– 3, in work begun prior to the events of September 11th 2001, the eighth paper for consideration [Gordon & Ford, 2002b] 'Cyberterrorism?'[27] stressed the importance of exploring not only the role of computer as target in so-called cyberterrorism, but as facilitator as well.

This work questioned the usage of the term cyberterrorism and suggested an urgent re-examination of the way in which we consider the convergence of terrorism with the current virtualization of many processes. For example, computers are now central to many parts of the communication infrastructure, providing the mechanics for everything from banking to power generation.

Thus, while damage to computers was once viewed as an entirely virtual attack, computer downtime now has a profound impact on the non-virtual world.

Essentially, this research examined terrorism broken down into an underlying matrix of eight different elements common to all terrorist events: perpetrator, place, action, tool, target, affiliation, motivation and outcome. Once accomplished, the effect of adding 'cyber' elements to each element of the matrix was considered separately.

---

[27] http://securityresponse.symantec.com/avcenter/reference/cyberterrorism.pdf

By taking such an approach, it was possible to reduce preconceived notions about the role of computers in terrorism and gain an appreciation of the many different terrorist-related abuses of information technology.

While the paper came to no firm conclusions with respect to cyberterrorism prevention, it did challenge even the validity of the term cyberterrorism, which tends to enforce a somewhat narrow view of the problem of terrorist equipped with computers. Furthermore, it argued strongly for the integration of computer expertise with more traditional defensive countermeasures, which should be designed horizontally (broadly) rather than vertically.

For example, consider handling so-called 'cyber attacks' on the national infrastructure. A vertical approach to such a problem is to create a specific organization dedicated to just monitoring, managing and analyzing attacks on the nation's Internet infrastructure. The advantage to such an approach is that deep knowledge of a particular attacker niche. Compare this to a horizontal approach. In such an approach, no new agency is formed; rather, computer skills and knowledge are integrated across existing arcas.

Contrasting the result of these different processes, the danger of the vertical approach is that it tends to compartmentalize our defense mechanisms. That is, an attack may involve elements across many different verticals. The attacks of 9-11 are an example of a somewhat non-traditional attack working extremely effectively. A more holistic approach to defense that was more integrative may have considered this broad but simple attack and blunted it.

Finally, the paper discussed the risks posed by large, badly managed clusters of connected machines. Just as one might restrict a hostile agent's access to explosives or guns, one can argue that the large stockpiles of connected machines pose a danger to the overall cyber-security of the Internet. It is therefore possible that a legal definition of a minimum standard of security should be constructing, providing at least a baseline for those managing large computer networks.

In addition to being selected to appear in *The International Library of Essays on Terrorism* [O'Day, 2004], this work was used by The White House staff as a tool to help policymakers understand the wider aspects and role of the computer in terrorism. It was presented at the IFIP World Computer Congress[28], Computer Security Institute Conference in Washington, D.C., and the European Institute for Computer Antivirus Research.

Privacy: Do As I Say, Not As I Do

Paper 9: Gordon, S. 2003. A Survey of Privacy Attitudes and Operational Behaviours in US, UK and EU Information Security Professionals. Keynote Presentation. From the Proceedings of the Compsec 2003 Conference. Queen Elizabeth II Conference Center. London, United Kingdom. Also available from: http://securityresponse.symantec.com/avcenter/reference/privacy.attitudes.behaviors.pdf

[28] http://www.ifip.tu-graz.ac.at/TC11/CONF/WCC2002/WCC_2002_Security_stream_brochure.pdf

Finally, *Privacy: A Study of Attitudes and Behaviors in US, UK and EU Information Security Professionals* [Gordon, 2003] is the ninth paper I submit. This research provided the first study quantifiably examining users' functional behaviors in a wide variety of computer security situations related to privacy and confidentiality[29]. This work is important as it explores the impact of human factors on the potential efficacy of technology as it relates to privacy.

The paper first examined historical and cultural aspects of privacy, before embarking upon a survey of technological threats to privacy online. In each case, reference was made to some form of technical remediation.

With these elements in place, a survey that measured the correlation between users' stated desire for privacy and their knowledge and use of privacy-enhancing technologies and actions was designed, and a pilot study launched in The United States. However, the pilot study proved to be flawed, due to lack of respondents lack of familiarity with the term P3P (Personal Privacy Platform) and confusion between P3P, PGP (Pretty Good Privacy) and 'Personal Privacy Policy'. The survey was reconstructed to address this validity issue.

Survey data was then gathered at UK, US and European security conferences and analyzed. Randomly selected IS professionals were queried anonymously regarding their

[29] http://securityresponse.symantec.com/avcenter/reference/privacy.attitudes.behaviors.pdf

desire to control the disclosure of information about themselves and/or their transactions, and their daily operational behaviours.

The data proved to be fairly simple to analyze, as the results were extremely clear. Within each population, a statistically significant disconnect existed between desire for privacy and the steps required to assure it. Apart from the numerical results, conversations with the subjects revealed that in many cases they were completely aware of the large gap between desired result and action.

According to [Festinger, 1957], when a person believes one thing but does another, a dissonance exists within the mind. Such dissonance is difficult to contain, and is therefore resolved in a number of different ways related to minimizing the negative consequences, focusing on the benefits of the action.

By showing that such a conflict exists, this research made possible more direct measurements of how this conflict is resolved; furthermore, by being aware of the potential conflict, methods were proposed to encourage users to resolve the dissonance in ways that help rather than hinder security.

After publication, this groundbreaking research on privacy issues was presented as the Keynote speech for Reed Elsevier's Compsec 2003 UK Conference; it is currently being used as a component of an awareness campaign for government and corporations worldwide.

Summary of Data Gathering Issues

In this section, several often-citied problems with sampling and survey techniques are considered and their relevance to my research considered.

Papers 2, 5, and 9 (*The Generic Virus Writer, Virus Writers: The End of the Innocence,* and *Privacy Cognitions and Behaviours in US, UK and EU Information Security Professionals*) summarized above rely heavily on data gathered by either directed or random survey. Sampling populations that engage in deviant behaviors presents a number of potential challenges. Even with some assurance of relative anonymity of such populations, subjects may be hesitant to discuss illegal or antisocial activities. Gathering a statistically significant sample can be difficult as the target population size is often unknown. Additionally, survey respondents that are self-selected present problem in terms of generalizing findings.

While the virus writer survey respondents in *The Generic Virus Writer* [Gordon, 1994b] were self-selected, they represented a relatively small target population; there were only two important virus writing groups at the time and responses were obtained from a statistically significant number of those individuals. Thus, responses could reasonably be generalized to the groups that formed the very foundation of the virus writing subculture.

While the respondents in this study were anonymous, due to my visibility in the community, issues with trust were overcome, and I was able to obtain extensive written, verbal and electronic data that both qualified the respondents and explored their actions and beliefs in depth.

In *Virus Writer: The End of the Innocence* [Gordon, 2000], the population of antivirus researchers, being extremely small and accessible, offered an excellent opportunity to obtain results that were representative of that community. On the other hand, gathering information from DEFCON respondents presented unique challenges; however, these were overcome by timing of the survey. This is further discussed in [Gordon, 2000].

In *Privacy Cognitions and Behaviours in US, UK and EU Information Security Professionals* [Gordon, 2003], the respondents were not self-selected; rather, a statistically significant number of respondents were randomly selected and individually queried at three computer security conferences. Respondents were pre-qualified by their attendance at computer security conferences and screen query as to their job function, to ensure they were currently working in the IT Security field at a technical level. As they were being asked questions that could represent their behaviors in negative ways, the survey was administered with anonymity.

Paper 8, *Cyberterrorism?* [Gordon, 2002b] presented some challenges related to gathering data related to terrorist activity. For example, some terrorist WWW sites forbid the use of their graphics or data without consent; yet, there was understandably no identifiable contact from which to obtain the consent. In such cases, if the information could be verified elsewhere, and was publicly available, it was incorporated into the study.

Reflections on Challenges

Mentorship and Collaboration

As this work was foundational, there were very few people with the desire or skill set to work collaboratively. I was fortunate to find two mentors, Dr. Harold Highland and Dr. Louise Yngstrom, both of whom had the skills and desire to see my work furthered; they provided much guidance, without which this research would not have been possible.

Additionally, my undergraduate mentor, Dr. Josh Tenenberg, offered much encouragement to me, sponsoring me for a University grant in the area of developing ethical curricula. However, due to geographical limitations, and the fact the Internet was not yet sufficiently developed to allow for communication using webcasts or netmeetings, the first several years of my work were done in relative isolation.

When I came to IBM's Thomas J. Watson Research Center, this limitation was overcome, as it is a multidisciplinary environment; however, there remained the limitation of the researchers' worldviews. In many (but fortunately not all) cases, while there were people with diverse skill-sets, they did not see the application of their skills to problems outside their area. Thus, worldview of researchers was a second limitation; I suspect this limitation exists still today, although significant progress has been made.

Resources

Resources were another limitation. Different resources are used, for example, when developing legal strategies for computer viruses than when developing solutions focused

on user education, or technology. Finding the information I needed was often difficult, as much of my early research was conducted in discipline-specific environments.

Integrating New Ideas into Culture

Generally, knowledge comes from within a discipline and begins to permeate culture, changing societal worldviews; for example, knowledge about HIV infection was developed by medical scientists and epidemiologists and was then integrated into educational outreach of public health programs; the effects of smoking were studied by physicians and researchers, and then integrated into elementary and high school health education programs.

However, in the case of computer viruses the needed changes *were* impossible to make from within the antivirus community. Impossible to impact from within, the only way to affect the necessary changes was to approach the industry from the outside, by facilitating dialogue that shaped the views of the world around the industry forcing the industry to follow. Thus, in this case, the changes took place from the outside.

Despite the challenges, it has been extremely rewarding. Since the inception of my work and my unrelenting focus on facilitating academic dialogue to foster considering of some focus on ethics and responsibility in the Computer Security field (which was the goal of the initial paper in 1994), the idea of ethics as part of a computer security curricula is gaining wide acceptance. For example, Purdue University [Purdue, 2004] established a multidisciplinary center in 1998, based on the recognition of the need for multidisciplinary approach.

Quoting Eugene Spafford, Director of CERIAS,

> "*The whole research and education program in CERIAS at Purdue is multidisciplinary in nature. We have linguists working with psychologists working with computer scientists who are working with economists. The results so far have been quite exciting, although we have sometimes had difficulty finding presentation venues where they understood what we have done. Our students clearly benefit from this mix, and we're trying to find ways to share the model with others. I am excited about what we are doing.*"

Other Universities now include this approach in their Computer Science/Computer Security curricula [Aycock, 2004; Holleran, 2004; Kabay, 2004; Chan, 2004; ECU, 2004; Ford, 2004].

General Approaches vs. Specific Approaches

One possible criticism of the work is that it is too generalist; that is, that especially in computer-related topics that the knowledge needed to really understand the implications of vulnerabilities requires highly specialized knowledge. There is some truth to this, and it is certainly possible to take a viewpoint that is too high-level. Thus, as is often the case in research, there is a delicate balance to maintain when viewing security problems from a broader perspective. By way of response, however, I believe that it is valuable to follow both a granular technical approach, and a more holistic generalist approach. Results from both approaches should be contrasted and compared. Any discrepancies should be carefully examined.

Another criticism of multi-disciplinary approaches to computer security is that analogies drawn from other areas may be misleading due to subtle but important differences in low-level operation. Critics might cite some of the conclusions drawn regarding computer monoculture, for example, concerning how biological inferences brought in to computing can lead to incorrect results.

This flaw of the interdisciplinary approach is entirely valid, and is something that must e considered when making inferences. As always, great care must be taken by the researcher to validate conclusions using as many different techniques are possible.

To the critic who would argue the need for specialization – I agree. However, within that specialization needs to be a broader worldview, especially for those making high-level decisions or doing research, to understand its application. Depth of knowledge can reduce problems and solve specific issues, but only by taking a step back and examining the system *in its entirety* can we ensure that researchers are actually working on the *right* problem.

Summary of Research Progress and Impact

This research demonstrates new and scholarly work in the use of non-traditional approaches to the computer security problem. Individually, these papers have laid groundwork that is being used to enhance understanding of computer security issues. Combined, they provide a solid base of peer-reviewed publications that illustrates the synthesis of computer security with other branches of science. Furthermore, their integration into the both the world of academia and our culture demonstrate they have

also gone a step further, and changed the way people worldwide think about computer security.

The initial work, Technologically Enabled Crime, set the stage for all the later work that I undertook. After first beginning with the process of fusing computer security with other disciplines, I decided to extend the base work to a broader audience. To do this, I followed several distinct paths, in order to illustrate the underlying research manifesto outlined in Technologically Enabled Crime.

In one thread, the virus writer work explored the nature and motivators of the adversary, and applied ethics, sociology and psychology to a domain that has traditionally been the realm of technologists. The results were both encouraging and useful: a more solid understanding of non-technical remediation to the virus problem was garnered, and the validity of a multi-disciplinary approach proven.

The virus analysis work performed on the first Macro virus (Concept) and the first Macro virus to target Excel (Laroux) continued to establish and cement my bona fides as a technical virus researcher. This was important both within the 'white hat' community of fellow virus researchers and within the cyclical virus writer community, as well as within the security community.

Continuing to emphasize the value of a broader approach, the systems theory paper looked at virus protection in its wider context. Once again, by moving beyond wholly technical solutions, and considering ideas in context, a pragmatic and effective approach toward virus protection could be derived. This way of thinking about the problem also

validated the underlying assumption that technology cannot solve security problems in a vacuum.

The concept of pragmatism and meaning was also touched upon in my work on testing. This work, accepted and adopted by governments worldwide, has forever changed the way that anti-virus products are tested. Vendors, testers, academics and the general public now think differently about antivirus software testing.

Finally, my work on cyberterrorism and privacy has also focused on the broader picture, synthesizing technical information and solutions with other disciplines. The privacy study demonstrates that even when technological solutions exist, they are frequently not deployed for non-technological reasons. Similarly, the research on cyberterrorism shows when facing an attack by an adversary, it is important to view the problem holistically not narrowly. This concept is vital to protecting the nation from computer-assisted threats.

Overall, the work has been well accepted, and ties together cohesively to powerfully demonstrate the power of examining technological problems from a far broader perspective.

Topics for Future Research

The discipline of computer security is ripe with opportunities for future research. Based upon the findings of my research to date, this future work should include:

- Continued research on individuals who write self-replicating programs, or participate in hacking activities; in particular, crossover between the two groups

at a societal level and a skill-set level, and connections with more organized criminal activity;

- Continued research into theoretical models for standards development, evolving into the areas of new uses and abuses of technology, i.e. such as spyware and adware;

- Designing college coursework focusing on holistic approaches to computer security issues. There is a shortage of available academic resource material in this area;

- Promoting multidisciplinary approach and collaboration through various methods including application for research grants in this area; publication in diverse journals and soliciting collaborative with academics in other fields.

- Creation of material for elementary school children, in collaboration with early childhood educators, focusing on the impact of virtual action in the 'real world'

- Research focused on communication methods and models used by security researchers, and the efficacy afforded by the various models

Future research: a personal perspective

Having been involved with computers for almost twenty years and having spent the last decade changing the way the world thinks about computer viruses and computer security, I want to mentor and supervise others who wish to undertake multidisciplinary research in these areas.

My current projects include acting as co-PI on a project with Purdue University and Symantec Corporation related to virus writers and hackers and co-PI on two pending National Science Foundation grant applications related to multidisciplinary studies of hacking and virus writers. This multi-disciplinary approach is beginning to be recognized as the only approach likely to yield long-term benefits. There is still much work to be done in this area, and I am actively taking part in doing it.

Ideally, now we will see others continuing the research I have started, especially in the area of virus writers. Currently there is no other credible scientific source of information about this subject, and I am hoping to mentor one or more students with an interest in this area. My deepest desire is to begin a transition that will result in passing this knowledge on. While my work is already being referenced as 'the' work in virus writing and hacker areas, I want to formalize both the materials and methods of teaching them over the next several years that will make this transition possible.

For this to happen there needs to be increased material available to educators. Thus, one area for future research is in developing curricula that explores the problem from many perspectives. Therefore, once I have achieved my PhD, I plan to work with others designing courses on malware that will, in addition to exploring technical aspect of malware, offer an integrated, multidisciplinary approach, teaching students not 'what' to think, but 'how' to think about computer security issues. Such an incorporation of my new work into formal curricula will continue to validate the legitimacy of the approach I have taken. In addition to lecturing at The University of Stockholm, The University of Hamburg, Indiana University, and Purdue University, I have recently been appointed to

serve on the Graduate Faculty at Florida Institute of Technology help guide students on projects related computer security research.

In the ideal world, educators will begin to expand on integration of ethical aspects of secure computing into daily lessons and life. Research that examines the ways in which young children synthesize computing ethics will be of great value and should be encouraged. Thus, in addition to the course developments described above, I have begun development of a book series for primary school students, creating educational materials that will instill a sense of reality in virtual interactions. The focus of series will be helping young children to understand there are real people on the other end of their Internet connection, and that their actions can have real consequences. My goal is to make sure this work is made available as widely as possible with the US and UK, and eventually, internationalized.

Finally, as technologies continue to develop, additional research and development of educational material that focuses on the human factors related to security technologies should be encouraged. Universities and technical colleges are encouraged to critically examine the structure of their academic curricula, and where lacking, to develop programs that integrate ethics, law, education, psychology and ethics. I hope to work collaborative with people involved in such development.

Conclusion

The changes that have come about as a result of the research presented in this document have been huge. Virus writers were once beheld as the denizens of the underground, solely bent on destruction, and best dealt with by technology in the best case, and law in

the worst case. This work has shown that in fact many young people who write viruses are much like other young people – and explained why the technical and legal approaches heralded by many two decades ago not only failed to solve the problem but in fact could contribute to making it worse.

Today, computer scientists who once scoffed at the idea of engaging in dialogue with young people who write self replicating programs are taking part in educational programs aimed at these young people. Technologists who saw no value in attending 'hacker conferences' such as DEFCON now appreciate and understand the tremendous opportunity observing such events can provide. This work has given us hope for the future, and a direction in which to focus our technical, legislative, psychological and educational efforts.

In other areas of security, such as 'cyberterrorism', my work expanded the boundaries commonly adhered to by organizations and individuals – boundaries that are not adhered to by the terrorists but which had heretofore been embraced by computer security professionals as 'playing by the rules'.

Prior to the release of this research, the concept of cyberterrorism discussed at security conferences and in academia was overly focused on computer as target. My research demonstrated that this is only one aspect of the inclusion of computing with terrorism. My development of a terrorism matrix provided a framework that allowed defenders to examine the impact of information technology as it intersected with every dimension of terrorist activity. By refusing to step on the 'cyberterrorism' bandwagon, but instead scientifically and methodically dissecting the elements of terrorist activity, this work

shone light in those dark places, and set new directions for research and practice, both for technologist and policymakers alike.

By scientifically and quantifiably showing that many information security professionals tend to adapt a 'Do as I say, not as I do' attitude about various technical issues related to privacy, this work has exposed the complacency and carelessness within human-technology systems that reside within critical infrastructures. It has opened the door for organizations to examine their own security cultures. It has also given technologists a direction for future research, finding ways to circumvent the need for 'human action' in some situations.

In summary, I have demonstrated that this work of the past decade has shone the light of scientific scrutiny upon stereotypes and the byproducts of premature consensus thinking[30] in many critical areas related to computer security. By introducing increased scientific scrutiny, it has aided in maturation of the discipline in ways which are both desireable and measurable. It represents new knowledge, and seminal work, and has changed the way people looked at the computer security problem overall. In essence, this work has changed the way in which the computer security discipline has developed and is developing.

Based upon the criteria of quantity, quality and scope, I have demonstrated that the body of work taken either in part or in whole is sufficient to earn the advanced degree. Specifically, in terms of quantity, I have included both papers listed for review, and

---

[30] Premature Consensus Thinking is also known as Groupthink [Festinger, 1957].

additional papers in the attached bibliography. These form only a subset of my work. In terms of quality, some are regarded as seminal work; foundational in their field and are required reading at Universities, and others are required reading in University Computer Science programs. Finally, in terms of scope, I believe that the work is both broad and well integrated.

Summarily, this research has done more than change the very way people think about computer security. It has provided a pathway for more integrative approaches to viewing security in a more holistic context. This will help prepare future generations to deal with the computer security problems we will face holistically, while at the same time contributing to the maturation of the computer security as a scientific discipline.

References

AAAS, 1993. *Ethical, Legal, and Technological Aspects of Network Use and Abuse.* Directorate for Science and Policy Programs. American Association for the Advancement of Science (AAAS) & Section of Science and Technology of the American Bar Association (ABA) Conference. Beckman Center. Irvine, California. http://www.aaas.org/spp/egii/welcome.htm

Aycock, J. and Barker, K. 2004. *Creating a Secure Computer Virus Laboratory (Case Study).* In U.E. Gattiker (Ed.), EICAR 2004 Conference CD-ROM: Best Paper Proceedings (ISBN: 87- 987271-6-8) 13 pages. Copenhagen:

Bandura, A. 1969. Principles of Behavior Modification. New York: Holt, Rinehart & Winston.

Bynum, 2003. Letter regarding Content of a Computer Ethics Course. Available from: http://www.southernct.edu/organizations/rccs/ethics_course.html

Cairns, P. and Thimbleby, H. *The Diversity and Ethics of HCI.* Available from: http://www.uclic.ucl.ac.uk/harold/ethics/tochiethics.pdf pp3.-15.

Chan, H. 2004. *Computer & Internet Security Management.* Course ISMT528. Hong Kong Business School. Available from http://www.bm.ust.hk/~ismt/course/syllabus/528_03Fall.pdf

Chess, 1998. In [Saarinen, J. 1998]. How to Protect Against Upconverting. Available from: http://www.infoworld.com/cgi-bin/displayTC.pl?/980601office97virus.htm

Cohen, F. 1986. *Computer Viruses.* PhD Dissertation. University of Southern California.

Dawkins, R. 1989. *The Selfish Gene.* Oxford University Press. ISBN: 0192860925

Denning, D. 1999. COSC 511 Schedule and Readings, Fall 1999. Department of Computer Science. Georgetown University. Also available from http://www.cs.georgetown.edu/~denning/cosc511/fall99/schedule.html

Dibbell, J. 1996. *Viruses are Good for you.* Wired 3.0.2. San Francisco, CA. Also available from: http://www.hnet.uci.edu/mposter/syllabi/readings/viruses.html

Draper, S. 2002. *PhDs by Publication.* Available from: http://www.psy.gla.ac.uk/~steve/resources/phd.html

ECU, 2004. *School of Computer and Information Science.* Internet and Computer Security Lab. Edith Cowen University. Perth, Australia. Available from http://www.scis.ecu.edu.au/research/icsl/index.asp

EICAR, 1998. *Where There's Smoke, There's Mirrors: The Truth About Trojan Horses on the Internet.* (Presentation based on research by Sarah Gordon & David M Chess. IBM Thomas J. Watson Research Center. High Integrity Computing Laboratory. Hawthorne, New York).

Festinger, L. 1957. *A theory of cognitive dissonance*, Stanford, CA: Stanford University Press

Fitzgerald, N. 2003. Available from: http://www.computerworld.com/securitytopics/security/virus/story/0,10801,85549,00.html

Ford, R. 2004a. *Introduction to Computer Security.* Private Communication with Professor Richard Ford (rford@fit.edu.).. Florida Institute of Technology. Melbourne, FL. Used with Permission.

Ford, R. 2004b, 2003. Microsoft, Monopolies & Migraines. *Virus Bulletin.* Pp. 9-11. December.

Geer, D., Pfleeger, C., Schneier, B., Quarterman, J, Metzger, P., Bace, R., & Gutmann, P., 2004. *CyberInsecurity: The Cost of Monopoly. How the Dominance of Microsoft's Products Poses a Risk to Security.* Computer & Communications Industry Association (CCIA). Available from http://www.ccianet.org/papers/cyberinsecurity.pdf

Gordon, S. 1994a. *Technologically enabled crime: shifting paradigms for the year 2000.* Computers and Security Journal. pp. 391-402. October. 1994. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/Crime.html

Gordon, S. 1994b. *The Generic Virus Writer.* From the Proceedings of the Fourth International Virus Bulletin Conference. Jersey, U.K. pp. 121-138. September 1994. Also available from http://www.research.ibm.com/antivirus/SciPapers/Gordon/GenericVirusWriter.html

Gordon, S. 1995a. *Virus Analysis: What a winword concept.* Virus Bulletin. September 1995. Also available from: http://www.virusbtn.com/magazine/archives/pdf/1995/199509.PDF

Gordon, S. 1995b. *Publication of Vulnerabilities and Tools.* From the Proceedings of the Twelfth World Conference on Computer Security, Audit and Control. Queen Elizabeth II Conference Center, Westminster, London, UK

Gordon, S. & Ford, R. 1995c. *Real-World Anti-Virus Product Reviews and Evaluation.* From the Proceedings of Security on the I-WAY. National Computer Security Association. Crystal City, Virginia.

Gordon, S. 1995d. *The antivirus strategy system.* From the Proceedings of the International Virus Bulletin Conference, Boston, MA. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/Strategy.html

Gordon, S. 1995e. *Social Engineering: Techniques and Prevention.* Computer Security. 1995

Gordon, S. 1996. Virus Analysis: *Excel.* Virus Bulletin. Also available from: http://www.virusbtn.com/magazine/archives/pdf/1996/199608.PDF

Gordon, S. 1996b. *Viruses on the Internet.* Virus Bulletin. pp. 14-17. August, 1996.

Gordon, s. 1997a. *What is Wild?* From the Proceedings of the 1997 National Information Systems Security Conference. National Institute of Standards and Technology.

Baltimore, MD. . Available from http://csrc.nist.gov/nissc/1997/proceedings/177.pdf

Gordon, S., Ford, R. & Wells, J. 1997. *Hoaxes and Hypes.* From the Proceedings of the International Virus Bulletin Conference. San Francisco, CA. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/HH.html

Gordon, S. 1998a. *The Worm Has Turned.* Virus Bulletin. July. pp10-12.

Gordon, S. 1998b. *Caught Red Handed.* Virus Bulletin. May. pp6-8.

Gordon, S. 1998c. Personal communication with anonymous colleague. Used with Permission

Gordon, S. & Ford, R. *1999. When worlds collide: information sharing for the antivirus and security community.* From the Proceedings of the International Virus Bulletin Conference. Vancouver, British Columbia. Also available from http://www.research.ibm.com/antivirus/SciPapers/Gordon/VB99/vb99final.html

Gordon, S. & Howard, F. 2000. *Antivirus software testing for the year 2000 and beyond.* From the Proceedings of the National Information System Security Conference. Washington, D.C. Also available from http://csrc.nist.gov/nissc/2000/proceedings/papers/038.pdf

Gordon, S. 2000. *Virus writers: end of the innocence.* From the Proceedings of the International Virus Bulletin Conference. Orlando, Florida. Also available from: http://www.research.ibm.com/antivirus/SciPapers/VB2000SG.htm

Gordon, S. 2002a. *Virus and Vulnerability Classification Schemes: Standards and Integration.* From the Proceedings of the Computer Security Institute Conference. Chicago, Illinois. Available from: http://securityresponse.symantec.com/avcenter/reference/virus.and.vulnerability.pdf

Gordon, S. & Ford, R. 2002b. *Cyberterrorism?* Computers & Security 21(7): pp. 636–647 2002. Also available from: http://www.compseconline.com/premium_article/premcs.htm#vol21issue72, and from http://securityresponse.symantec.com/avcenter/reference/cyberterrorism.pdf

Gordon, S. 2003. *A survey of privacy attitudes and operational behaviours in US, UK and EU Information Security Professionals.* Keynote Presentation. From the Proceedings of the Compsec 2003 Conference. Queen Elizabeth II Center. London, United Kingdom. Also available from: http://securityresponse.symantec.com/avcenter/reference/privacy.attitudes.behaviors.pdf

Guralnik, D. 1976. *Webster's New World Dictionary of the American Language.* Second College Edition. Cleveland, OH. William Collins and World Publishing.

HEX, 1992. *Forty Hex.* 12, 6. Available from: http://www.etext.org/Zines/ASCII/40hex/40hex-12.006

Holleran, J. 2004. Personal conversation. Used with Permission.

Highland, H. 1990. *Computer Virus Handbook*. Elsevier Press. Oxford, UK. ISBN 0-946395-46-2

Highland, H. 1995. Private Communication.

IFIP, 1994. International Federation for Information Processing. Technical Committee 11. Available from: http://www.ifip.tu-graz.ac.at/TC11/CONF/SEC94/

IFIP, 2002. *E-Government and Cybterrorism.*. International Federation for Information Processing. Montreal, Quebec, CA. Available from http://www.ifip.tu-graz.ac.at/TC11/CONF/WCC2002/WCC_2002_Security_stream_brochure.pdf

Kabay, M. 2001. *Studies and Surveys of Computer Crime.* Course Document 1.1.2. Norwich University. Norwich, CT. Also available from http://www2.norwich.edu/mkabay/methodology/crirne_studies.htm

Kaminski, J. 2004. Available from: http://eicar.weburb.dk/app/Schedule/ViewSession?id=131&database=eicarprogram

Kaspersky, E. 2004. Available from http://www.kasperskylab.nl/virusanalyst.html

Kephart, J. & White, S. 1993. *Measuring and Modeling Computer Virus Prevalence.* From the Proceedings of the 1999 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California. pp.2-14.

Kephart, J., & White, S. 1991. *Directed-graph epidemiological models of computer viruses.* From the Proceedings of the 1991 Computer Society Symposium on Research in Security and Privacy, pp. 343–359.

Kohlberg, L. 1984. Essays on Moral Development: The Psychology of Moral Development (Vol. 11). San Francisco: Harper and Row.

Kotadia, M. 2004. *Bulk of PC's Infections Pinned to One Man.* Retrieved from the WWW on August 11, 2004. Available from: http://news.com.com/Bulk+of+year's+PC+infections+pinned+to+one+man/2100-7349_3-5287664.html

Kuhn, T., 1970. *The Structure of Scientific Revolution.* 2nd edition, ed. University of Chicago Press, 1970. Chicago, Illinois

Lee, A. & and Harley, D. *Back to the Future: Fresh Approaches to Malware Management.* From the Best Paper Proceedings of the 2002 European Institute for Computer Antivirus Research Conference. Berlin, Germany pp. 76-108

Ludwig, M. 1997. *Computer Viruses, Artificial Life, and Evolution.* American Eagle Publications. ISBN 0–929408–07–1

Middleton, J. 2001. *Anna virus writer offered IT job.* VNU News Center. Available from: http://www.vnunet.com/News/1117945

Moore, D., Paxon, V., Savage, S., Shannon, C., Standiford, S., and Weaver, N. 2003. Inside the Slammer Worm. *IEEE Security and Privacy.* Available from http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html

Morley, P. 2001. Processing Virus Collections. From the Proceedings of the Virus Bulletin Conference. September, 2001. Prague. Also available from http://www.nai.com/common/media/vil/pdf/pmorley_VB_conf_2001.pdf

O'Day, A. 2004. Cyberterrorism? In *The International Library of Essays on Terrorism.* Ed. Alan O'Day. University of Oxford. ISBN 0 7546 2426 9.

Peterson, P. 1995. Virus-L Digest.Available from http://www.control.auc.dk/~magnus/Mailboxe/firewall-archive/0587.html

Purdue, 2004. *Threats from Malicious Software.* Center for Education and Research Information Assurance Lafayette, Indiana. Also available from: http://www.cs.purdue.edu/homes/mja/426PowerPoints/Malicious_Software.ppt

Raiu, C. 2004. Available from http://www.kasperskylab.nl/virusanalyst.html

Rogerson, S. 1996. *The Ethics of Computing: The First and Second Generations.* The UK Business Ethics Network News. In *The Stanford Encyclopedia of Philosophy (Winter 2001 Edition)*, Edward N. Zalta (ed.). Also available from: http://plato.stanford.edu/archives/win2001/entries/ethics-computer/

Rogerson, S. and Bynum, T. 1995. *Cyberspace: The Ethical Frontier.* Times Higher Education Supplement. The London Times. In *The Stanford Encyclopedia of*

Philosophy (Winter 2001 Edition), Edward N. Zalta (ed.). Also available from http://plato.stanford.edu/archives/win2001/entries/ethics-computer/

Rusch, J. 2002. *The Social Psychology of Computer Viruses and Worms.* From the Proceedings of the INET 2002 Conference. Crystal City, Virginia. Also available from http://inet2002.org/CD-ROM/lu65rw2n/papers/g10-c.pdf

Saarinen, J. 1998. *How to Protect Against Upconversion.* Infoworld. Available from http://www.infoworld.com/cgi-bin/displayTC.pl?/980601office97virus.htm

Schofield, J. 2003. *Bringing Internet to Schools Effectively.* v8, 3. ed. William Peters. Global Issues. United States Department of State.

Shipley, G. 2002. *Maximum Security.* Sams Publishing. ISBN 067231871

Slade, R. 2003. *Software Forensics: Collecting Evidence from the Scene of a Digital Crime.* McGraw Hill Publishers. ISBN 0071428046

Stucker, H. 1997. *Scans: Among the Virus Thugs.* Wired Magazine. Issue 5.04. Also available from http://www.wired.com/wired/archive/5.04/scans.html?pg=7

Taylor, S. 2004. Private communication. Used with Permission.

Tippett, P. 2000. in Poulsen, K. 2000. Lawmakers Slam Antivirus Biz. Available from: http://www.securityfocus.com/news/32

White, S. 1998. *Open Problems in Computer Virus Research.* From the Proceedings of the 1998 Virus Bulletin Conference. Munich, Germany.

Yang, A. 2002. *Computer Security and Impact on Computer Science Education.* Indiana University of Pennsylvania. CCSC Northeastern Conference. Also available from: http://palms.ee.princeton.edu/fiskiran/repository/p233–yang.pdf

Yngstrm, L. 1996. *A systemic–holistic approach to academic programmes in IT security.* Stockholm University/Royal Institute of Technology, Report series No. 96–021, 1SSN 1101–8526.

Websters, 2004. Websters Collegiate Dictionary. Eleventh Edition. Springfield, MA.

Appendix 1: List of work submitted as part of this application

Gordon, S. 1994. *Technologically enabled crime: shifting paradigms for the year 2000.* Computers and Security Journal. October. 1994. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/Crime.html

Gordon, S. 1994. *The Generic Virus Writer.* From the Proceedings of the Fourth International Virus Bulletin Conference. Jersey, U.K. September 1994. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/GenericVirusWriter.html

Gordon, S. 1995. *Virus Analysis: What a winword concept.* Virus Bulletin. September 1995. Also available from: http://www.virusbtn.com/magazine/archives/pdf/1995/199509.PDF

Gordon, S. 1996. Virus Analysis: *Excel.* Virus Bulletin. Also available from: http://www.virusbtn.com/magazine/archives/pdf/1996/199608.PDF

Gordon, S. 2000. *Virus Writers: End of the Innocence.* From the Proceedings of the International Virus Bulletin Conference. Orlando, Florida. Also available from: http://www.research.ibm.com/antivirus/SciPapers/VB2000SG.htm

Gordon, S. 1995. *The Antivirus Strategy System.* From the Proceedings of the International Virus Bulletin Conference, Boston, MA. Also available from: http://www.research.ibm.com/antivirus/SciPapers/Gordon/Strategy.html

Gordon, 1997. *What is Wild?* From the Proceedings of the 1997 National Systems Security Conference. National Institute of Standards and Technology. Baltimore, MD. . Available from http://csrc.nist.gov/nissc/1997/proceedings/177.pdf

Gordon, S. & Ford, R. 2002. *Cyberterrorism?* Computers & Security 21(7): pp. 636–647 2002. Also available from: http://www.compseconline.com/premium_article/premcs.htm#vol21issue72, and from http://securityresponse.symantec.com/avcenter/reference/cyberterrorism.pdf

Gordon, S. 2003. *A survey of privacy attitudes and operational behaviours in US, UK and EU Information Security Professionals*. From the Proceedings of the Compsec 2003 Conference. Queen Elizabeth II Center. London, United Kingdom. Also available from: http://securityresponse.symantec.com/avcenter/reference/privacy.attitudes.behaviors.pdf

Appendix 2: Quotes related to this research

- *Computer security expert Sarah Gordon is to the virus-writing underground what anthropologist Clifford Geertz was to the Balinese. Gordon has been quietly studying people who write computer viruses to understand more about their ethical development, perceptions of themselves and the world around them, and their motivations for releasing viruses into the wild. (Wired, April 1997).*

- *The acknowledged authority in this area is Sarah Gordon, who has written extensively in this area and in related ethical areas. (Shipley, 2002).*

- *Sarah Gordon's credentials as an antivirus expert, one adept at dealing with the lethal creations of young hackers, are impeccable. She spent years debugging her own personal computers while she worked as a juvenile crisis counselor. Since 1997 she has worked at the preeminent antivirus lab in the country, IBM's Thomas J. Watson Research Center, in Hawthorne, New York. (Forbes, April 1998).*

- *"They need to be educated to see that viruses are having a real impact, and that people get hurt. They need to know that self-replication is an interesting computer concept, but it's not a science, and it's not cool." (IEEE Voices, March 1998.*

- *Sarah Gordon is the world expert in the area, but even she admits: "I can't give you a simple answer to the question 'What sort of people do this?' If I could, we could develop a generic approach to solving the problem. The reality is that this is an extremely complex issue..." . Her research has made her something of a myth in virus circles. (The Guardian, July 1999)*

- *In her definitive works on the subject, The Generic Virus Writer II, she gives a number of different reasons including: "relief from boredom, actively seeking fame, exploration, malice and peer pressure". (The Guardian, July 1999)*

- *Gordon began her research in 1994 and updated "The Generic Virus Writer," her study of four virus authors, two years later. She investigated about 100 people, verifying their information through follow-up interviews with family, friends and other sources ... "As ethical standards and notions of right and wrong mature, most of the young writers grow out of their misbehavior and are replaced by other youngsters, she said. For that reason, the community continues to reinvent the wheel rather than evolve." (Government Computing News, August, 1999)*

- *"The image of the virus writer as an angry social malcontent bent on destruction is generally wrong," Gordon says. "Most – especially the teenagers – code for thrills and are often disconnected from the reality of what their creations can do," she says. "They don't believe that their code can actually hurt anyone," Gordon says. "It's actually a normal level of ethical development for their age group," she adds. (PC World, 2000)*

- *Gordon is an anomaly in the anti-virus industry... Sarah has spent years interviewing, profiling and exchanging ideas with virus writers, and in turn, they talk to her and trust her. (CNN 2000)*

- *Studying the psychology of virus writers and hacker... Her research at Symantec includes a focus on ethics and technology. (PBS Frontline 2000)*

- *Through her research, Gordon has learned that virus creators are usually not mean, immoral people who set out to ruin people's days. "(They're) mostly within the normal ethical range for their age. They just don't understand the impact on the real world of their actions," Gordon said. "Sometimes their grandfathers or parents get a (computer) virus and (the virus writer) totally flip-flops and immediately says, 'This is wrong,' " she said. (San Francisco Chronicle, 2001)*

- *Contrary to popular myth, Gordon says, cyber-rebels aren't underground loners, and they're not necessarily nerdy – or even smart. She believes they join 'the*

dark side' of the Internet because they don't extend the same moral code from the real world to the virtual world. She blames teachers, journalists and parents for the breach. (CNET 2002)

- *While thousands of researchers toil to thwart the creations of virus writers, very little has been done to investigate who these shadowy figures are and why they do what they do. Much of it is the work of Sarah Gordon. (USA Today, 2002)*

- *"How long might it take to develop a moral code that is consistent from the physical to virtual worlds? It doesn't happen in one generation. It will take a long time. But we have to do something about it because the shift won't happen automatically. Educators can start teaching kids at a very, very young age what things are acceptable and what aren't" (CNET, 2002)*

- *"People who study science need a multidisciplinary approach. If you like computer code, get involved in computer science courses, but get involved in something else, too: Get a degree in engineering or biology and then get an internship at Symantec or IBM Research. Find what you love and just do it. Find out what makes your heart beat fast, and run with it." (CNET, 2002)*

- *"Teaching ethics to young children is more than just teaching then 'do's' and 'don'ts'. While that is important, the crucial thing is to teach how to make decisions, and how to apply that decision making in various environments and circumstances including environments and circumstances with which some teachers (and many parents) have little familiarity. It is extremely complicated; we are still defining what is acceptable in virtual environments and interactions, and until we have a clearer understanding and agreement societally, it is difficult to instill those principles in young children." (RAE Internet, 2003)*

Appendix 3: Television, radio and magazine appearances as expert on social and technical aspects of computer security.

These selected articles demonstrate the breadth, and depth of my multidisciplinary research, and illustrate the result of my concerted effort to shift the media perception of technical, social and ethical implications of technology.

CNBC: The Street. 2004. *Searching for Andy.* Technical Forensics and the MyDoom virus author. Available as on–site video presentation only (copyrighted).

BBC News, 2003. *A glimpse inside the virus writer.* Research on virus writers featured on BBC News. http://news.bbc.co.uk/2/low/technology/3240901.stm

Silicon, 2003. *Virus writing hackers are the biggest threat.* Article referencing research on virus writers and hackers. http://www.silicon.com/software/security/0,39024655,39116705,00.htm

CSO Magazine, 2003. *Don't let your babies grow up to be hackers.* Research on virus writers and hackers featured in CSO Magazine. http://www.csoonline.com/read/070103/briefing_babies.html

IDG Net, 2003. *Virus Experts Debate Bug Names.* http://enterprisesecurity.symantec.com/content.cfm?articleid=2957&EID=0 Quoted on issue of virus naming.

ZDNet UK, 2003. *The hacker challenge.* Quoted on research related to lack of convergence between hacking and virus writing communities. http://insight.zdnet.co.uk/communications/networks/0,39020427,2133479,00.htm

Computing, 2003. *Security experts pay scant attention to privacy issues.* http://www.webactivemagazine.co.uk/News/1147423

C–NET. 2002. *Deciphering the hacker myth.* Profiled for research with virus writers and hackers. http://news.com.com/2008–1082–829812.html?tag=pt.salon

Microsoft B–Central. *Hacking into the mind of a hacker.* Research on hackers utilized by Microsoft publication. http://bcentral.com/articles/enbysk/164.asp

Microsoft B–Central. *7 things to know about virus writers.* Research on virus writers in interview with Microsoft publication. http://www.bcentral.com/articles/enbysk/160.asp

PBS Frontline. 2002. *Studying the psychology of virus writers and hackers.* Feature and profile on PBS for research on virus writers and hackers. http://www.pbs.org/wgbh/pages/frontline/shows/hackers/whoare/psycho.html

MSN Techs and Gadgets, 2002. *Hacking's not just for geeks anymore.* Interviewed by Microsoft for research on hackers and virus writers. http://msn.com.com/2100-1105-937343.html

Financial Review. 2002. *The lure of cyber larceny crosses the gender divide.* Research on virus writers and gender issues quoted. http://afr.com/specialreports/report2/2002/07/25/FFX2PSMQY3D.html

USA Today. 2001. *Hot on the trail of virus writers.* Research on virus writers featured in this profile. Full page feature. http://www.usatoday.com/tech/news/2001-05-07-virus-tracker.htm

Fast Company. 2001. *Livin' la vida Boca.* Featured (with husband) in article on career paths and life-choices. . http://www.fastcompany.com/online/46/boca.html

Computer Reseller News. 2001. *Delving into the online underworld.* Research on virus writers featured. http://www.cm.com/Components/Search/Article.asp?ArticleID=23672

San Francisco Chronicle. 2001. *Germ warfare battle against viruses escalate.* Quoted for research in psychological and technical aspects of information warfare. http://www.sfgate.com/cgi-bin/article.cgi?file=/chronicle/archive/2001/05/28/BU56977.DTL

Konrad, R. 2001. *Deciperhing the Hacker Myth.* ZDNet Australia. Available from http://www.zdnet.com.au/news/security/0,2000061744,20263321,00.htm

CNN, 2000. *Hacking into the minds of virus writers.* Profiled for research with virus writers and hacking. http://www.cnn.com/2001/TECH/science/02/10/index.virus/index.html

WIRED. *A worm writers worst friend.* 2001. Profiled for research on virus writers and psychological profiling .http://www.wired.com/news/women/0,1540,43839,00.html

IDG News, Boston. *Why there aren't good viruses.* Quoted on technical non–viability of 'good' viral code, and problems related to medical analogy. http://www.virusmd.com/aboutus/idg/defcon.html

Time Digital. 2000. *The New Hot Zone.* Featured in article on computer viruses for research done with the Massively Distributed Systems Group, Antivirus R&D Team, IBM Thomas J. Watson Research Center.

Government Security: *Technology Solutions in Defense of the Homeland.* Featured in book 'Tangled Web'. Review: Access Controls and Computer Systems. http://govtsecurity.securitysolutions.com/ar/security_bookshelf/

Computer Crime: *Phreaks, Spies and Salami Slicers.* 2000. Profiled in book about computer crime for ethics and technology research. April, 2000.

UK ITV News. *Profiling virus writers.* Research on virus writers featured. http://www.itn.co.uk/c4news/home/20000707/Story04.htm

PC World, 2000. *What makes Johnny and Jane write viruses?* Research on virus writers and gender issues featured. http://www.pcworld.com/news/article.asp?aid=34405

Government Computing News. 1999. *Profiler analyses the enemy.* Research with virus writers featured. http://www.gcn.com/vol18_no27/com/429–1.html

WIRED, 1999. *Inside the virus writers mind.* Feature article on research done with Massively Distributed Systems Group, Antivirus R&D Team, IBM Thomas J. Watson Research Center . http://www.wired.com/news/politics/0,1283,20624,00.html

NBC Evening News. June 1999. Interviewed by NBC News, evening edition, for research on Virus Writers done with Massively Distributed Systems Group, Antivirus R&D Team, IBM Thomas J. Watson Research Center.

Rolling Stone.1999. *Notes from the Virus Underground.* Quoted by authors and virus writers in special feature on psychology of virus writers and virus writing community.

Frankfurt Museum of Art, 1999. *The Love Bug.* Research on virus writers featured in Frankfurt Museum of Art. http://www.dvara.net/HK/iloveyou.asp

Forbes, 1998. *At work.* Profiled for research on virus writers for Massively Distributed Systems Group, Antivirus R&D Team, IBM Thomas J. Watson Research Center. http://www.forbes.com/asap/1998/0406/018.html

Voices, 1998. IEEE Voices. Photographed and quoted by IEEE publication. http://www.eetimes.com/news/online/online99.html

Facts on File: Career Ideas for Kids Who Like Adventure. 1998. Chosen as example of 'Computer Security Expert' in publication exploring career options for children. http://www.amazon.com/exec/obidos/tg/detail/-/0816043221/104-6698306-5460703?v=glance

WIRED. 1997. *Among the virus thugs.* Profiled for research with virus writers. http://www.wired.com/wired/archive/5.04/scans.html?pg=7

WIRED, 1997. *Heart of Darkness.* Interviewed for research with Bulgarian virus writing community. http://www.wired.com/wired/archive/5.11/heartof.html

Dedication

For Beulah

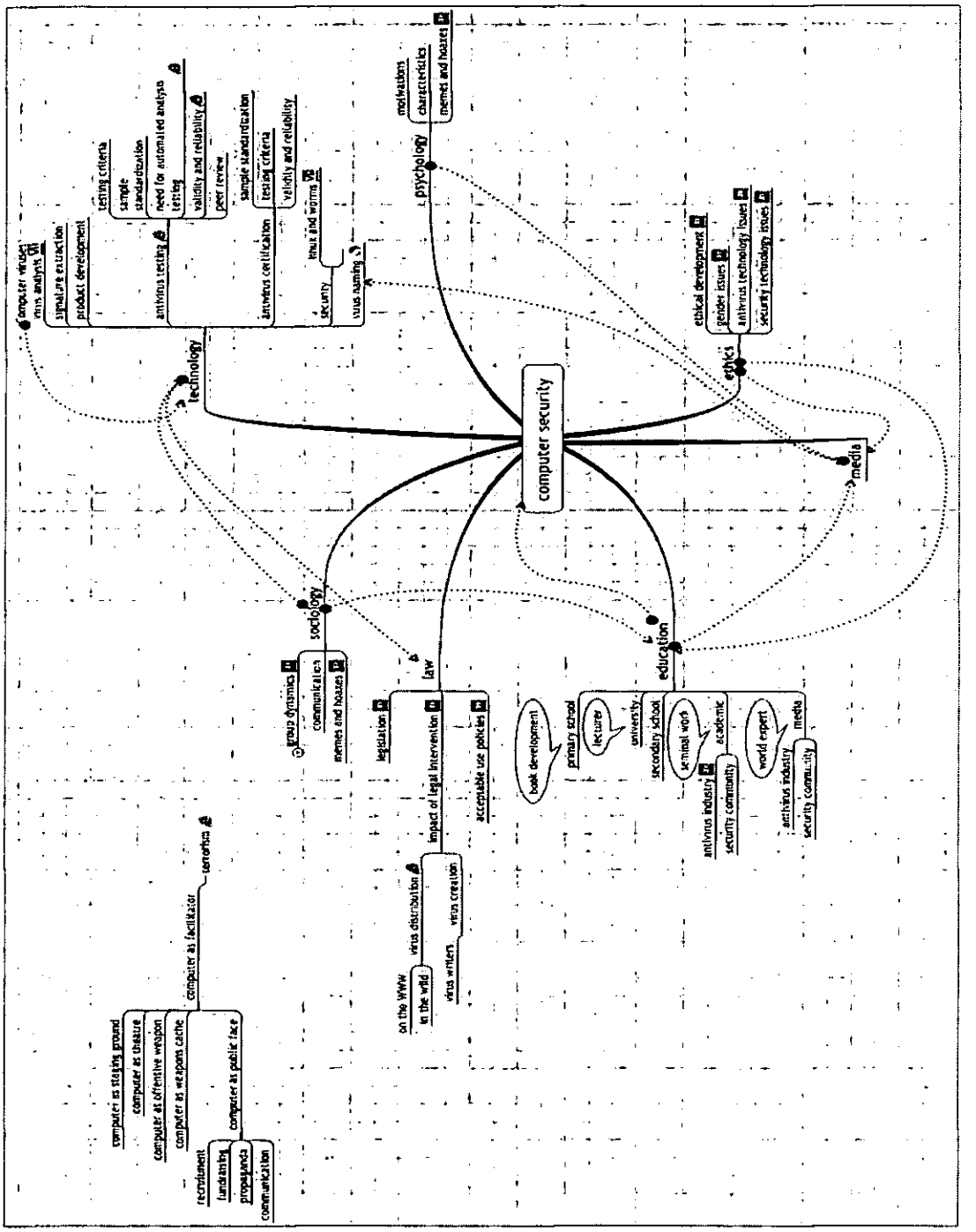


Diagram 2 (Enlarged

Technologically Enabled Crime: Shifting Paradigms for the Year 2000

By Sarah Gordon
E-mail:sgordon@low-level.format.com

"Best Paper & Presentation", Internation Federation for Information Processing Sec 94 Technical Committee 11, Curacao, Netherlands, Antilles, May 1994.

- Abstract
1. Introduction
   - 1.1 Traditional epidemiological studies
   - 1.2 Social Aspects
2. The causal connection
   - 2.1 Malicious software
   - 2.2 Individuals
3. Epidemiology
   - 3.1 How viruses programs travel
     - 3.1.1 Virus Exchange Networks
     - 3.1.3 Virus Distribution Sites
     - 3.1.4 Virus Distribution Robots and File Servers
     - 3.1.5 Virus Instruction Books
     - 3.1.6 Viruses for Sale
   - 3.2 How hacking tools travel
     - 3.2.1 Private BBS
     - 3.2.2 Networked BBS
     - 3.2.3 Usenet
     - 3.2.4 FTP Sites
4. Social Factors
   - 4.1 Cyberspace as facilitator
   - 4.2 Technology as enabler
5. Future Trends
6. Solutions

    - 6.1 Laws
    - 6.2 Ethical Considerations
7. Conclusion

- Bibliography
- About the Author

Abstract

This paper will consider the social and ethical factors involved in the transmission of computer viruses and other malicious software. In addition to the people, we will consider the part the systems and technology play in the spread of this sort of data. We will draw parallels with one of the more well known scientific paradigms, the medical one, and note the similarities with the problems we now face. We will describe the evolution of methods of virus distribution: virus exchange bulletin boards, virus exchange networks, distribution sites, robots/servers, and books. The paper will discuss viruses for sale and make some comparisons between distribution of computer viruses and the distribution methods of ``hacking tools''. Other issues examined in this paper include the characteristics of individuals involved in the distribution of these types of programs, and problems of legal redress, as well as possible solutions based on ethics and ethical theory.

Return to Top

Introduction

We have available today a global system of communication technology. There also exist programs whose purpose is to disrupt the way this system functions. Moreover, the system is the perfect medium to host and transfer the very programs designed to destroy the functionality of the system itself. In this paper we will discuss the factors usually neglected in studies conceming computer virus infections.

Return to Top

1.1 Traditional Epidemiological Studies

```
ep-i-de-mi-ol-o-gy \,ep-e-,de^--me^--'a^:l-e-je^-, -,dem-
e^--\ n
[LL epidemia + ISV -logy] (ca. 1864)

1: a branch of medical science that deals with the
incidence, distribution,
and control of disease in a population

2: the sum of the factors controlling the presence or
absence of a
disease or pathogen (Webster's)
```

There are various factors commonly considered when estimating the probability of virus infections. We have factors such as the ability of the virus to replicate, the amount of contact any given machine has with the general population of computers, and the presence of any computers currently infected. Elaborate studies have been done to calculate the possibilities of any given population becoming infected. In one such study by Dr. Alan Solomon [Solomon, 1990], one conclusion is that early detection is a very effective way to reduce the incidence of viruses in a population of computers. In fact, early detection is cited as one of the crucial factors in limiting infection. One such model [see Footnote 1] illustrates how finding a virus contributes to its detection and eradication.

There are cases however, where a virus being `found' means it will spread further and further; the same can be said of some hacking tools. These cases are where the malicious programs are `found' on computing systems, where they have been placed for exchange or distribution. These are programs which will not be detected in their `current state' by any virus detector or casual search methodologies. When they are found, by people looking for them (and in some cases by the casual observer who just happens to see them, download or ftp them, and use them), they spread from user to user and their use becomes widespread; in some cases, epidemic.

Return to Top

1.1.2 Social aspects

In addition to being concerned with detecting viruses which are active in computing systems, we now find ourselves in the position of needing to detect and identify viruses and other malicious software which are non-active. We are faced today with an entire system of communication technology which is the perfect medium to host and transfer the very programs designed to destroy the functionality of the systems. We suggest that technologies not only tend be created out of human endeavor and the accompanying social values, but to shape the values of the communities from which they arise; that they can take on an ethical/moral dynamic of their own. These values, as we will show, are not always consistent with the values of the communities which create them.

Return to Top

2. The causal connection

In this section, we will examine the sorts of programs which are sometimes used in criminal and/or unethical ways. People which make use of the current technology to distribute the tools and information will be discussed.

2.1 Malicious software

By malicious program, we refer to a program designed to perform a harmful action. This action could range from deliberate destruction of data, as is the case with some viruses, to the interception of confidential information, as is the case with programs such as the recently publicised sunsniffer. For the purposes of this paper, the computing technologies referred to are those which are affected, or which have the potential to be affected.

While it is not required for a program to do obvious damage to classify as a virus, for the purpose of this paper we will stipulate a virus as a program that replicates in some environment, alters executable code and does damage by controlling your computer system without your knowledge or consent; we will stipulate a trojan as a program which appears legitimate, but which does deliberate damage to your computer system's files. While viruses have for the most part been confined to personal computers running under MS-DOS, we are beginning to see both more interest and more viruses written for UNIX based systems.

The hacking tools discussed are computer programs including trojanized login programs, which capture passwords, shell scripts which exploit operating system bugs and text files which give instructions on how to hack computer systems.

Of course, these programs alone do no damage. They must be installed, executed or read and used as "instruction manuals"; this is accomplished initially by a human. It is interesting to note that many people insist that programs are `unethical'. Other voices insist the programs are not capable of being ethical or unethical; they are simply code. Traditionally, programs were not seen as capable of being ethical or non-ethical in and of themselves, primarily because they were not autonomous agents. However, viruses have the capability to be exactly this. For this reason, if the viruses we are seeing today are in any way the precursors to full-scale autonomous agents, we should be concerned with which ethical models we will incorporate into them. Will they make their decisions based on the good of all of society; will they make their decision based on unwavering moral principles? Will they be totally self-preservationist? There appears to be little if anything to indicate these programs with which we are concerned in this paper bear any relationship to artificial intelligence or artificial life despite claims to the contrary by their producers, and for this reason are not ethical or unethical in and of themselves.

Return to Top

2.2. Individuals

The sort of people which play a role in the distribution of this malicious information vary. There are malicious, intentional players, as well as non-malicious accidental players. We will begin with the virus writers. It would be an error to place them all in one category. They are as diverse as their viruses; each with his own motivation and each subscribing

to his own choice of distribution method. The term `his' is specifically used because there is no evidence of any female virus writer who participates consistently in distribution of computer viruses. The gender issue is one which is discussed in the paper The Generic Virus Writer [Gordon, 94]; it will not be disussed further at this time except to note there is a gender issue.

Virus writers can come from all walks of life; they are diverse in age, location, academic background, and goal. In some cases, the goal is malicious in nature; in other cases, there appears to be no malicious intent. The same is true of the hacker. The traditional profile of hacker [Swanson, Chamelin and Territo, 92] as young adult male, 19-25, socially inept seems to be somewhat inaccurate. There are women involved in the hacker culture, not just as `fans' and `hangers-on', but as contributory entities.

Another similarity between types of the virus-involved individuals and roles of the individuals in their subculture and that of hackers and those involved in their subculture is that both may exhibit `parasitic' behaviour. Parasite in this context refers to people who have no skills of writing replicating code, nor any abilities related to what is commonly referred to as *hacking*. These people participate in the culture by helping distribute the programs, and the information in crude, traditional ways; telephone conversations, bulletin board chats, uploading/downloading files on dial-up bulletin boards; use of the Internet in some cases to transfer files, and maintenance of huge repositories of information which they cannot contribute to, but which they can allow others to `benefit' from. They feed off of the `work' of others. For this reason, they are often referred to as `parasite hackers' or `parasites' by members of their social communities.

These are not the only people involved in the epidemiology of malicious programs. Commercial software companies are involved. At least 64 instances of DOS-based commercial software have been released with infected files or infected boot sectors. There are increasing numbers of reports of infections on commercial and shareware CDs released for DOS based machines [Footnote 2]. Innocent users are sometimes carriers. We are all familiar with the sneaker net mode of infection. where an office worker carries a disk to his/her co-worker, and in transferring the files or booting from the shared disk, also sometimes transfers the virus. Users can also transfer viruses by not following proper procedures in their environments; not taking the virus threat seriously. Anti-virus software is often disabled by users because it is too slow or not installed at all because the installation is considered too complex. When this lack of provision for detection exists, the user can play host and distributor to viruses without ever being aware of their existence. Administrators also sometimes play a role in the distribution of viruses and other malicious programs, unknowingly. This will be discussed further under section 3.1.3 Virus Distribution Sites.

Return to Top

3. Epidemiology

Having defined some types of programs that are used to cause disruption and criminal activity in our networks, aspects of cyberspace and technological development which can contribute to the problem and the general characteristics of some of the people involved, we will now look at the methods by which the people distribute the programs and information.

3.1 How virus programs travel

Viruses are exchanged and distributed via at least six methods. The first, the virus exchange BBS, is perhaps the most well known. We will trace the growth of viruses as a novelty, to the beginnings of their place in commercial ventures. To discuss the motivations of the persons involved in each of these individual steps is beyond the scope of this paper. We will answer the questions: how are the machines and the technology used as methods of communicating information; what kind of information is being communicated?

Return to Top

3.1.1 Virus Exchange BBS

One of the common methods utilized by intentional computer virus distributors is the virus exchange bulletin board. The bulletin boards are similar in most respects to mainstream bulletin am bulletin board systems. The sotems. The software used by the indivby the individual system operators varies. Mrs varies. Many of the systems aresystems are accessible via teliphone, and some are accessible through telnet. From a humble beginning in Sofia, Bulgaria (the site of the first known virus exchange system), virus exchange bulletin boards have grown into big operaem was operated by Todor Todorov in Sofia Bulgaria; it made viruses available initially on an `exchange' basis, but later offered the viruses to anyone who cared to take them. In its initial stage, it encouraged the creation of new viruses by requiring the upload of a new virus in exchange for access to any and all viruses. The system had a total of 294 users and was used primarily by local callers. The number of "regular" files on this system was at least double the number of viruses; according to the system operator, the non-virus files were the most frequently accessed. Following the popularization of this system via negative publicity as well as ``word of mouth advertising" by users, other systems began to emerge. Currently, virus exchange bulletin boards are known to exist in North America, Latin America, Europe (including Switzerland where it has become a crime to offer viruses via a BBS; and Holland, where it is also a criminal offense); Australia, Asia and Africa. The systems sometimes state they are Virus Research Bulletin Boards. Some of the systems are "private"; others allow access to anyone who wishes to participate. These individual systems have led to a new development; that of the virus exchange network.

Return to Top

3.1.2 Virus Exchange Networks

These systems were for the most part well-publicized by word of mouth, electronic mail and advertising on other systems of the same type. While hack/phreak systems had been in existence for some time, the virus exchange phenomenon was a relative latecomer to the underground scene. Within roughly a three year period, the operators and users of such systems had formed a relatively small but tightly knit community, and the formation of organized networks followed. The networks provided even faster distribution of new viruses to network members. The majority of these systems operated using regular dial-up modems and a network structure similar to the Fidonet. The networks have names such as vX-Net (Virus Exchange Net), NuKEnet (named after the NuKE virus writing group which founded the network), and MeltNet (an exclusive net which has never been known to release a virus outside of the network). These networks have been observed to overlap; often systems will participate in more than one of the networks. In some cases, the networks will publicly identify themselves as "Virus Research BBS", while in another network they are known by their virus exchange system or virus distribution affiliated name. One such instance was the Virginia Institute of Virus Research, which was also known as the Black Axis BBS. This system was represented in the Fidonet echomail conference as a virus research center; it was identified in another network as the world headquarters for the NuKE virus writing group, operating under the name "The Black Axis". This is not an isolated instance, but is perhaps the most well known. The virus exchange systems as exist via regular dial-up access are easily accessible to users. Since they are self-administered, they are not usually subject to any form of external review or assessment.

Return to Top

3.1.3 Virus Distribution Sites

As interest in viruses grew, the abilities and resources of the virus writers and distributors grew. Some of the young virus writers became college aged; access to internet facilities became available. Internet virus sites became more commonplace, and information about the ever-changing locations was transferred at the same fast rate as the viruses themselves. It is not uncommon to find university ftp sites used as virus distribution sites. This creates a problem for overworked administrators, who in many cases have no idea what is passing through their systems. How can we detect these viruses? In some cases they are not directly detectable, having been encoded by some standard (or non-standard) utility such as uuencode; in other cases they are archived. Both these methods make their detection by current scanning methodologies difficult if not impossible. They are not active in memory, or existing in any form which a traditional scanner may recognize. In many cases these are MS-DOS viruses, which are transferred using UNIX machines. They are often in and out of sites before most administrators know their systems have

been used for the purpose of holding or transferring the data.

Return to Top

3.1.4 Virus Distribution Robots and File Servers

Use of automated distribution programs known as bots and servers is a relatively recent addition to the methods used to distribute viruses. By contacting one of the servers via electronic mail, or by asking the `robot' for the files, a user can relatively anonymously retrieve viruses via the internet. The connection can of course be monitored, but they do not appear to be routinely monitored by the administrators or by the users themselves. One recently programmed file server reportedly transferred to users approximately 15,000 to 20,000 files (viruses and text files) per week during its three months of operations. There were approximately 1000 files available for download/transfer from this server. The operator of the server learned to make and use bots during his self-taught experience with the Linux operating system. Following the success of the server, he programmed a bot which was actively distributing viruses on the Internet Relay Chat. He states he put the server online to do something that had never been done before -- internet wide virus distribution. As the server was anonymous, there is no way to know what sort of users accessed the files, their intended purpose, or the result of the accessibility.

According to the server operator, the supplier of internet service declared a breach of contract following the huge volume of file transfers; he was forced to remove the server. Such servers, and bots, can be used for distribution of any type file, not just viruses; this transfer of information can be accomplished with relative anonymity.

Return to Top

3.1.5 Virus Instruction Books

Books on how to make viruses have become popular, and contests are sponsored to build the smallest virus; the most politically incorrect virus; the virus best able to defeat anti-virus programs. In 1990, Mark Ludwig copyrighted The Little Black Book of Computer Viruses. This book contained general information about types of viruses. It contained computer source code for the viruses as well as an order blank readers could use to order the code on disk; it also contained what the book refers to as "compiled executable programs for all of the viruses and related programs in this book". There was a disclaimer, requiring the purchaser to assume full responsibility for any damage that may be caused by any of the programs. The viruses themselves were not particularly innovative. Several of them have been found in the wild since the publication of the book. This book created some controversy, followed by the release of a second book. The second book was released without much attention in the United States; however, in France, there was considerable controversy surrounding the release of the book. The final ruling of the French court on distribution of this book is not known at this time. There

have been other books published which contain computer virus source code. They have not achieved the notoriety of the Ludwig book. We are not suggesting any books should be banned. However, there are ethical considerations with which computing professionals need to be concerned. We will discuss these further later in this paper.

Return to Top

3.1.6 Viruses for Sale

Viruses are offered for sale by individuals. Several such offers were posted in various Fido and Usenet newsgroups. In addition, some magazines carry advertisements for viruses. Magazines also offer virus source code; the sale of these magazines appears to be legal at this time in the United States. Virus writers and distributors have begun creating and selling new viruses to some anti-virus product developers for inclusion in the 'scanner' programs. Government and industry sources have been said to purchase or obtain viruses from virus exchange systems or virus distributors, to perform testing of the anti-virus software they are considering. The virus phenomenon has become big business.

Return to Top

3.2 How hacking tools travel

Hacking tools, such as shell scripts which exploit system holes, buglists, etc. appear to travel via different sorts of paths.

In the case of these tools, and the people who exchange them, the scenario appears to alter slightly. The majority of hacking tools appear to be created after the announcement of a software bug. Hackers then create tools to exploit the bugs. In some cases, the hackers themselves find the bugs. There appears to be more creativity, individual action, and intentional sharing of the information among hackers than among the virus involved individuals; however, the information has tended to be limited to those who are judged (within the subculture) of understanding and contributing to further development of the tools. In some case, individuals obtain one set of tools and use them to obtain others by simply taking them from the filesystems of the tool developers.

Primarily they have been shared amongst individuals in the relatively tightly knit hacking community, until recently. We are now beginning to observe a shift which is cause for concern:

- Hackers sharing programs ---> Hackers sharing programs
- Shared among a small group ---> Shared among small groups
- Not widely distributed ---> Distributed more widely; wide-banded;
- Not generally used maliciously ---> Used maliciously;

This shift can be observed by following the distribution of one hacking tool commonly know as the sunsniffer. Initially the sniffer was distributed only to a very few people. The source code and executable code for this sniffer were recently '"widebanded". Widebanding refers to indiscriminate intentional distribution of a program, through every available method. In some cases this is done to make tracing of the original distributor more difficult.

The sniffer, which compromised the security of large number of systems on the internet, worked by using a feature of the operating system called /dev/nit. This is the network interface tap, and it can read/write from/to different interfaces. The program was configured to place /dev/nit in promiscuous mode, because it could then read all traffic from any machine on the cable, even routed mail. Administrators who had not properly configured their own /dev/nit helped enable the compromise of their own systems. However, this "hole" was designed into the system, making this compromise possible. It is not feasible to disable a machine to prevent its compromise.

As people became more aware of the use of this program by a few individuals, the potential for apprehension of the individuals increased, so the tool was distributed a bit more widely. At the same time, other individuals began to find this "sniffer" on machines which had been compromised; they would then take a copy of it to use elsewhere. Copies of the sunsniffer were placed on publicly available FTP sites, where any user with access to anonymous FTP could obtain the program. The shift we are observing whereby hackers are distributing information such as this on a much wider scale than before is illustrated by the speed and manner of the distribution of this sniffer.

What has brought about this shift? As suggested earlier, technology can bring about an ethic of its own that is not necessarily in keeping with the ethic of the creators of the technology. While this can be said of virtually any technology, it appears to be particularly applicable in the case of computing technologies. This will be further discussed in section 5., Future Trends, in which we will examine some of the reasons for the shifts we are observing.

Recently, there have been more hacker voices calling for public dissemination of both operating system holes and fixes. There are diversified opinions in both communities regarding whether or not such information distribution would benefit either of the communities in regards to their respective goals. Whether or not this idea gains widespread acceptance in either community remains to be seen.

Return to Top

3.2.1 Private BBS

While private BBS are set up, offering some tools, these tools tend to be of relatively minor significance: war-dialers, phreaking information, information easily available

about operating systems. Some BBS do contain more technically advanced materials, but access to them appears to be more exclusive than is the case with virus exchange bulletin board systems. Most of the information on h/p/a/v (hacking, phreaking, anarchy and virus) systems is of lower quality; most of the tools found are said to be trivial.

Return to Top

3.2.2 Networked BBS

Networked systems seem to be much less frequent, and those that do exist do not appear to offer the more exclusive tools.

Return to Top

3.2.3 Usenet

An interesting aspect of hacking tools is the use of Usenet news for their distribution. Source code for hacking tools appears on various Usenet groups, but usually this is after hackers have had access to them for some time. Such source code can be saved by readers, and compiled to create tools such as shell scripts to install port hoppers, and so on; It has been our experience in talking with a number of persons who have arrived relatively recently into the `hacking scene' that they are not capable of using these tools. The problem usually appears to be the necessity to modify the programs for different platforms; these people simply do not possess the ability to do it. Another problem is the apriori technical knowledge required. It does little good for a hacker to have a device that exploits a bug in kmem, for instance, if he does not know what to do once he has access to kmem. Simple programs for altering utmp files require modification as simple as directory paths; frequently, people do not have even the skills to do this. Commonly, such persons will access a UNIX system and enter DOS commands such as DIR, or type HELP.

This is not to say that the tools are not useful in helping them to learn; however, it is clear that these tools require more than a casual knowledge of the systems they are intended for use on. As the toolkits become more developed, less skill is required on the part of the users. However, some basic knowledge is still required.

Return to Top

3.2.4 FTP Sites

The use of Usenet for distribution of such tools is not the only way the Internet is used to facilitate the travel of hacking tools. FTP sites are routinely used for drop sites. These in many cases require special accesses or passwords, but in some cases tools are left on public sites, either through oversight on the part of the individuals involved, or

intentionally.

Return to Top

4. Social Factors

The connection between certain aspects of current computing technology and the crimes/activities being facilitated will be examined, with emphasis on the paradigm shifts which have been proven to improve the overall health of other forms of scientific research.

4.1 Cyberspace as facilitator

We will now consider the aspect of this cyberspace environment known as dehumanization. Not all computing technologies are heavily influenced by the dehumanization and other psychological aspects of `cyberspace' which are seen in the environment surrounding the `malicious computer program', but it should not surprise us that people who have little contact with other human beings due to their intense immersion in the electronic communities we have designed have lost sight of their humanity. It follows that the impact of their actions is often seen, at least by them, as impacting machines, not other human beings.

We should also consider the aspects of cyberspace which facilitate inequality, and the possible results of these inequalities. This environment is no different than in any other aspect of society; it is normal for people to be unequal. For example, we do not all have access to the same quality of health care; not everyone has even a house in which to put a terminal. Cyberspace however, introduces a unique form of inequality in that the sort of information which is becoming available will provide what could be a very extreme advantage to those who `have' versus those who `have not' -- indeed, this advantage/disadvantage could impact the electronic community in such a way that the community could become unable to maintain itself entirely. Unequal access to information puts those who do not have the access at the distinct disadvantage of ever being able to fulfill their potential in the electronic society. While this is inherent in most societies, we are in a position now which could enable us to minimalize some aspects of social inequality by careful planning and policy making. Unlike other areas, this structure is not yet intact; there is still time to integrate equalizing factors. Most importantly, we need to consider what sorts of information belong in cyberspace; what sort of access policies should govemments envision; is the idea of access for everyone feasible or even desirable.

At this time, cyberspace does tend to facilitate some inequality; this inequality is manifested in the number of `victims'. It can be argued that there is a great equalization, due to lack of real world visual biases or clues inherent in net communication and interaction; however, it is important to consider that along with the lack of the visual

'bias' triggers comes a lack of contextual clues. Without these clues, often people do not realise their behaviour is unacceptable. If it is alright to do one little thing, another little thing is added to it. Eventually, you can end up with a very anti-social behavior, which was totally acceptable every step of the way by one's peer group. This is not to suggest that we should find a way to take real-time, real-space clues and integrate them into net societies. As users are given more and more power, the potential for trickery, lies, deceit and abuse increases right along with the potential for 'good'. It may be wise to consider the nature of cyber-societies and the processes of social influence within them. [Sproull, 93]

Return to Top

4.2 Technology as enabler

In addition to the people, we must consider the part the systems and technology play in the spread of this sort of data. We can best do this by drawing a parallel with one of the more well known scientific paradigms, noting the similarities with the problem we now face:

Medical Science in the early 1960s:

- We can do it
- We should do it
- We must do it

Communication Technology Today:

- We can do it
- We should do it
- We must do it

The ":it" in the first case refers to advances in medicine relating to health care, and research; in particular fields such as genetic engineering. What occurred during this time was a remarkable advancement of technology which left scientists and researchers in somewhat of a quandary over exactly what, and how much, of this research and development should be put into common usage or pursued at all. We find a similar situation today, with computing technologies not only surpassing the abilities of administrators and users to understand them, but of the technologies themselves at times enabling their own destruction. It is perhaps wise to consider at some point what safeguards we should require. In the 60's, science turned to the field of ethics -- a field which was dying according to some -- and asked the question "Just what exactly should we do? What is -right- to do?". From this introspection, the field of bio-ethics emerged. [Bartels, Smith, 93] [Gustafson, 70]

When we look at medical science, and medical research today, we find questions being asked:

The Medical Science Paradigm today :

- We can do it
- Should we do it?
- How should we do it?

We can observe the shifts resulting from the interaction with ethical concerns. This shift has meant perhaps less scientific 'advancement', but perhaps has placed medical science more in line with its true goals. The same could be said for integration of ethics with other scientific disciplines. As the technologies of computing today advance, they tend to focus on what the machines can do. In this assumption, we could be neglecting what we really need and want them to do.

Return to Top

5. Future Trends

The technologies described to this point which have enabled the sorts of crimes we are now seeing in our global computing environments were surely not created or designed to facilitate these sorts of behaviours. We must, however, take a serious look at contributory factors.

It could be the case that we have simply allowed technology to progress too quickly, with insufficient planning. This is not to suggest that we should stifle technology, but that we may need to begin now to pay particular attention to the ethical model that the technological model is generating. As an example, consider FSP and FTP applications. We have seen how FTP (File Transfer Protocol via connection state protocol) can in some cases allow files to be transferred anonymously. This is a good and necessary thing, and its potential for abuse or misuse could be minimized by correct configuration policies. FSP, or File Server Protocol (Transfers via Connection list) in which you have a connection only during pings, requests, etc. are an improvement in that you do not tie up resources during inactivity; however, use of FSP usually requires no special privileges to set up and no special ports; it doesn't require separate file systems, and anyone can set up this sort of 'server'. We are seeing the same sorts of problems with these FSP servers as we are seeing with the DCC (Direct Client to Client transfer services) applications and Bots that are being used to transfer viruses and other programs on IRC (Internet Relay Chat).

The anonymity of both of these applications plays a role in the ethical models of behaviour that have developed around their uses. While FTP sites are used to transfer the sorts of programs and information with which we are concerned, there appears to be a

much higher incidence of FSP sites being used on a regular basis to transfer this information and data. The controversy surrounding anonymity and pseudo-anonymity is one which will probably continue for a long time as we learn the effects of such freedoms. However, what we can see now is that these sorts of anonymous applications do provide almost a "Use Me for Your Own Purposes" sign.

Other technologies which have had huge influence on society have developed relatively slowly, enabling us to at least somewhat predict future trends; however, in the case of computing technology, not only do we have few precedents on which to build our analysis, the technology by nature is rather esoteric. This creates an environment perfectly adapted to the development of pseudo-revolutionary counter culture and the exploitation of those who have, or are perceived to have, power. Additionally, the trends which we are able to predict would seem to indicate that legal methods of redress are inadequate at best. A proactive approach to the problems facing us as relating to hacking, virus writing/distribution and dissemination of information which has the deliberate design of being used in a harmful or malicious way would have to include ethics and education. The types of ethics and education will be discussed briefly in the next section, Solutions.

Return to Top

6. Solutions

Both legal and ethical solutions to some of the problems discussed in this paper are worth considering. However, both have limitations, and need to be used in a cooperative, or multidisciplinary approach. We will look now at some of the methods we can use to address the problems.

6.1 Laws

Laws are one method. There are however, problems with laws addressing computer viruses, virus source code, and hacking `tools'. As evidenced by the recent cases involving members of a well known `hacker' group, jurisdiction can be a problem. In one particular case, the alleged perpetrator physically resided in the United States; the system he reportedly attacked was located in Australia. The question of jurisdiction has, to this point, made prosecution impossible. [Cook, 93]

Laws concerning viruses have problems due to their lack of enforceability, jurisdiction and the matter of recovery. As we have shown, the nature of the methods of exchanging computer viruses and hacking tools tend to hamper any real assessment of exactly how much information is being exchanged and by whom. While of course there are ample mechanisms for monitoring information exchanges, we need to be concerned with various policies (both legal and ethical) when we consider monitoring communications to ensure their `acceptability'. The vast majority of known virus writers are not capable of

providing recovery should they actually be convicted of a crime, successfully prosecuted, and found guilty. Finally, there is the international nature of virus distribution, which adds to the already complicated situation.

While courts have usually found that information distributors are not strictly liable for damage caused by distribution of misinformation, recent decisions have held that distributors of products can be held strictly liable for the results of reliance on misinformation contained in the product (Cook, 93). The United States Commerce Department, in January 1990, found that International system administrators have an affirmative obligation to review the contents of their systems to locate improper or illegal traffic, specifically traffic in programs which have controlled export under the Export Administration Act or the Arms Export Control Act. While laws are still evolving and no one knows for sure what the end result will be, it seems safe to assume that administrators and commercial system owners will eventually face possible liabilities for actions of their users, such as virus infected products, viruses distributed via networks, stolen credit card information transferred via their networks, users businesses disrupted because adequate safeguards were not in place. This however does not solve the problem. The administrators may have a responsibility ethically and perhaps eventually legally to know what is going on on their systems; however, we cannot ignore the obvious gap between what a system should enforce and what it is actually expected to enforce. We must also be cognizant of the gap between what we can expect will be enforced the social policies and mores that exist in any given environment [Neumann, 93].

The concept of Free Speech as a Constitutional Right is invoked by many proponents of unrestricted virus "exchange" in the United States. There are forms of speech that are not protected by the First Amendment to the United States Constitution; additionally, there are precedents which bring serious questions to the First Amendment defense. The virus problem is not confined to the United States alone, and any laws specific to any individual country may not be applicable in another country. The discussion of free speech and/or First Amendment rights is beyond the scope of this paper; it is mentioned due to its large role in the defense of virus writing in the United States.

Finally, we may wish to examine ways in which laws can be used to create positive ethical models in individuals and groups. First, quoting a release from the Technical and General Assemblies of the International Federation for Information Processing [see Footnote 3]]

> In view of the potentially serious and even fatal consequences of the introduction of `virus' programs into computer systems, the Technical and General Assemblies of IFIP urge:
>
> 1. all computer professionals to recognize the disastrous potential of computer viruses;
> 2. all computer educators to impress upon their students the dangers

of virus programs.

3. all publishers to refrain from publication of the details of actual virus programs;

We see a very good suggestion as to how we may begin to positively influence students and young people. We can observe how this has been seen to work in the past by looking at the issue of drinking and driving. At one point in time, drinking and driving was a personal issue. As we as a society began to see some of the consequences of this interaction, we began to pass laws which restricted the such behaviour. There was some resistance to this type of law initially, which people saw as an infringement on their right to drink alcohol and drive their vehicles. However, as the law became more widely accepted, people began to refuse to drink and drive on the principle that it is `wrong' to do. Policymakers and lawmakers are very aware of this form of societal control. However, they are often not very aware of the societal structure of `cyberspace', and for this reason there is the danger that laws they make will not create the desired ethical model, but will instead create a backlash or revolutionary movement against the society. By continuing to take time to develop realistic policies and effective laws, it is possible we can avoid such a backlash.

Return to Top

6.2 Ethical considerations

The ethical approach to addressing these concerns is one worth further consideration. What role does ethics currently play in our computing environments? What role, if any, should it play? Ethics is quite the `in' word, and is often promoted as the be-all and end-all solution to all the problems we face dealing with virus and malicious software distribution. Ethics, however, cannot and should not be seen as a `behaviour regulator'. It is not a drug one can force down someone's throat, and cure them of their "disease". If we are to use ethics to help us to solve some of the problems discussed in this paper, where and how should we begin? There are several areas of immediate concern.

Commonly, ethics is promoted, if at all, in our computing environments as something related to individual action. While ethics certainly can be important in matters of our interpersonal actions and subsequently on our actions as they impact the society, we seem to ignore the issues of ethical evaluation of institutions (Ladd, 93).

Questions related to distributive justice (here, I refer to rights in the sense of both negative and positive rights; specifically, what can I expect to do free from any infringement from government or individuals, and what duty does my society have to provide me with access, freedoms, security, development and distribution of resources), and other ethics of management are worthy of consideration.

There have been voices calling for more clearly defined professional ethics and more

involvement of professional societies in defining and promoting `professional ethics'. Considering ethics is by nature a reflective, critical field, it would seem that while ethical norms may be documented, to assume we can arrive at some `ethical statement of principle' is somewhat unrealistic. Ethics are not laws, rules, policies or agreements. It is not something one can put on from the outside. Of course, ethics can and should play a role in creation of codes of conduct. Such codes of conduct are necessary and important tools in imparting behavioural guidelines to others [Forrester, Morrison 94]. We must be careful not to confuse codes of conduct, which are based on ethical principles, with ethics themselves. If we do not take care, we are subject to a slippery slope where we may believe that we are somehow `above' the ethical principles we apply to others. This can create a hypocrisy which only exacerbates the problems that are created by other factors, as outlined in this paper. The development of codes of behaviour is often looked to as one ethical solution. This may be a factor in showing individuals what is acceptable, but cannot be viewed as a method for instilling ethical behaviour in any group.

Another concern is what type of "ethics" should we look to for help in understanding and solving the problems of malicious program distribution. Is it the ethical theory itself that we must reintegrate into the educational system? According to the ACM/IEEE-CS Curriculum Task Force, undergraduate programs need to "prepare students to understand the field of computing both as an academic discipline and as a profession within the context of a larger society". One of the main goals is cited as exposing students to the "ethical and societal issues that are associated with the computing field." The question of whether this instruction should consist of ethical theory or application is prominent. One school of thought is that we need to teach ethical applications now, before the problem gets any worse. Another view is that teaching ethical theory will allow us to develop ethical applications which will continue to develop as the technology develops.

Return to Top

7. Conclusion

When a new technology emerges , a paradigm associated with that technology appears or is borrowed from an associated technology. As the technology develops towards maturity, the paradigm shapes its development. At certain points, it becomes apparent that the paradigm is no longer appropriate, and a paradigm shift occurs. Typically this is first seen as an outlandish if not heretical move by some maverick individual. But if the shift is appropriate, it becomes adopted by the scientific community, and then serves to shape or even control the further development of the technology. Without such paradigm shifts, the technology may become stagnated, or even dangerously out of touch with its aims and the society around it. Computer science is no exception.

I have argued above that we are now at the point where a significant paradigm shift is necessary in this area. The speed with which global electronic communication is developing has brought with it an enormous benefit to all those fortunate enough to be

able to exploit it. It has also brought opportunities to those who are willing to abuse it. The way in which it has introduced relative and absolute anonymity to its users itself may encourage acts which would otherwise have appeared to be too risky to the perpetrator. That is, its very nature may encourage various kinds of antisocial activities, ranging from innocent pranks through serious malicious damage to data and individuals to downright criminal fraud. The speed and power of the technology itself enables these activities to take place, and encourages them. Since its principle users are relatively young, and may be impressionable or unprincipled, an ethos has developed in which it is `cool' to be an outlaw. Moreover, the inherent power embodied in being able to control the `system' is itself potentially irresistibly attractive.

It is natural, given the way that societies tend to develop, that antisocial or otherwise undesirable activities lead to legislation against them, designed to contain or eradicate them. This is the point we have reached with such excesses on the Internet. This is the current paradigm of control, and the one that is influencing the development of the technology. However, legislation is notorious for not solving the problems it is designed to deal with. A paradigm shift is now necessary, both in the way the technology develops further and in the way that malicious activities associated with it are combatted. The problem of internet abuse cannot be solved by trying to legislate it out of existence. It is necessary to promote an ethical approach to computing. This itself requires there to be an ethical model of developing computer science. The paradigm for this technology can no longer be determined purely along scientific lines. Introducing ethics into the way the technology is used will help to instill appropriate ethics in the users of the technology, and thus to reduce the numbers of abusers. If this program is successful, it will soon sound outdated and even `lame' to say "it's ok to do it if it isn't illegal", just as it has become `uncool' to drink and drive; not merely illegal, but unethical, and not the sort of thing that enchances the image and status of a potential role model.

We cannot eliminate the social aspects of malicious computer program development and distribution through solely legal means, or through solely technical means. We can look to technology for detection in some cases, and to law for prosecution or relief in some cases. In all cases, resources to enable us to emphasise and integrate ethical computing behaviours in all areas -- not just in areas relating to viruses and hacking -- may provide a stablizing influence. Our computing environments are very vulnerable regarding distribution of information -- after all, it is what they were designed to do. I suggest that we need to focus somewhat more on what we were designed to do: to behave as rational self-policing beings and to impart this ethical model to people learning the technology. Without the proper interaction of laws, education and ethical development, there is a very real risk that this technology will soon become unusable and ultimately self-destructive.

Return to Top

Bibliography

1. Bartels, Smith, 93 New Frontiers in Genetic Testing and Screening: The Human Genome Project, Bartels, Dianne M. and Truesdell-Smith, Elizabeth, Center for Biomedical Ethics, University of Minnesota, August 1993
2. Cook, 93, "Network Traffic Liability: 1993", Cook, William J., op-ed for AAAS Invitational Conference on Technical, Ethical and Legal Aspects of Computer and Network Use and Abuse. report forthcoming.
3. Forrester, Morrison 94, Computer Ethics: Cautionary Tales and Ethical Dilemmas in Computing, MIT Press, 1994
4. Gordon, 94 "The Generic Virus Writer", Gordon, Sara (in progress, Virus Bulletin Conference)
5. Gustafson, 1970 "Basic Issues in the Biomedical Fields", Soundings 53, Summer 1970 151ff
6. Ladd, 93, "Critical Reflections on Ethical Issues Relating to Computer and Network Use and Abuse", Ladd, John, Dept. of Philosophy, Brown University. AAAS Invitational Conference on Technical, Ethical and Legal Aspects of Computer and Network Use and Abuse. report forthcoming.
7. Neumann, 93 "Limitations of Computer-Communications Technology", AAAS Invitational Conference on Legal, Ethical and Technological Aspects of Computer and Network Use and Abuse. report forthcoming.
8. President's Commission for the Study of Ethical Problems in Medicine and Biomedical and Behavioural Research, "Splicing Life". U.S. Government Printing Office
9. Solomon, 90 Epidemiology and Computer Viruses, Solomon, Alan, 1990, S&S International
10. Sproull, 93 "Social Influence in Electronic Groups", Sproull, Lee, December 1993 from "Atheism, Sex, and Databases", Sproull, Lee and Faraj, Samer, in progress -- Presented at AAAS Invitational Conference on Technical, Ethical and Legal Aspects of Computer and Network Use and Abuse. report forthcoming.
11. Swanson, Chamelin and Territo, 92, "Criminal Investigation", Swanson, Charles, Chamelin, Neil and Territo, Leonard, ed. Butcher, Phillip A. and Rosenberg, Elaine. pp. 53

Return to Top

Footnotes

1. Solomon model: In the Solomon model, the rate of new infections is proportional to the number of infected PCs, to the number of uninfected PCs and to the probability of infection. The rate of infections being eradicated is proportional to the number of infected PCs, and to the probability of detection.
2. A list of viruses distributed with commercial software, compiled from VIRUS-L, RISKS-FORUM and other public sources, identifies virus infections transmitted through either commercial or government entities in which the distributor would generally have been considered to be a "reputable source". Incidents which were

unwilling to fully disclose, or incidents in which the source of the infection was unsure were omitted. This list was obtained from Wallace Hale of the PCVRF. It is noted that any addition information may be requested from, or forwarded to cmcdonal@wsmr-emh34.army.mil.

3. "The resolution was formulated by the chairman of IFIPs Technical Committee TC-11 `Computer Security', Professor William J. Caelli, of Queensland University, Brisbane/Australia, and the chairman-elect of IFIPs TC-9 `Computer and Society', Prof. Klaus Brunnstein of Hamburg University. IFIP General assembly asked the president, Ashley Goldsworthy, to inform all member societies and to ask the governments to take proper actions." (Used with permission)

I am grateful to Tim Martin, Jon David, and Harold Highland for their comments on an earlier draft. They are not responsible for any errors or ommissions.

Return to Top

About the Author

Sarah Gordon's work in various areas of IT Security can be found profiled in various publications including the New York Times, Computer Security Journal and Virus Bulletin. She is a frequent speaker at such diverse conferences as those sponsored by NSA/NIST/NCSC and DEFCON. Recently appointed to the Wildlist Board of Directors, she is actively involved in the development of anti-virus software test criteria and methods. She may be reached as sgordon@low-level.format.com

Return to Top

back to index

The Generic Virus Writer

By: Sarah Gordon
E-mail:sgordon@low-level.format.com

- Abstract
- Introduction
- The Generic Virus Writer
- Ethical Models
  - Kohlberg's model
- Gender Issues
  - Gilligan's Model
- Methodology
- Virus Writers
  - The Adolescent
  - The College Student
  - The Adult
  - The Ex-Virus writer
- Conclusion
- Bibliography
- About the Author

Abstract

This paper presents four case studies of individuals involved in virus writing. The research was conducted by using surveys, and by conducting interviews via e-mail (electronic mail), electronic chat and in-person sessions. Ethnographic and demographic data were collected, as well as information relating to how the individuals view their relationships to their peers and to society in general. Some data relating to cognitive reasoning abilities was collected. This data was used to examine the individuals' moral development in light of ethical and moral developmental models based on the research of Lawrence Kohlberg. Gender based issues in virus writing are examined using the model developed by Gilligan [1].

Return to Top

Introduction

In any area of scientific investigation, there is the danger of overgeneralisation and stereotyping. In the case of virus writers, one manifestation of this danger is that of assuming that there is some homogeneous group of people who write viruses, and that it is possible to talk about the psychology of 'the' virus writer. In reality, there are different types of virus writers, each with his own nature, circumstances, skills and ambitions. This paper will not attempt to be all-inclusive; it will exami ne three 'types' of virus writers by using case studies of individuals who fit into these categories:

```
(a) the young adolescent individual
(b) the college student
(c) the adult/professionally employed individual.
```

We will try to shed some light on the differences in their make-up, and thus to assess the difference in the nature of any danger posed by each of them. If the virus writing population is not as homogeneous as some may assume it is, then monolithic solutions to 'the problem' (such as blanket or overkill legislation, certain forms of ethical solutions) may well be much less effective than is being assumed in certain quarters. We will observe differences in how they think, how they operate, and in how they view the rest of the world.

We will also look at the ways in which people may progress through these classes. This progression will lead us to a fourth category:

```
(d) the mature reformed ex-writer of viruses.
```

While the last category is often ignored (since the apparent threat is gone) it needs to be considered with as much care as the other three types. Not only are people of this fourth category potentially very skilled technically in the defence of cyberspace against members of the other three categories, but they also represent the kind of people into which we hope members of the other three categories will develop.

The virus writer has been characterized by some as a bad, evil, depraved, maniac; terrorist, technopathic, genius gone mad, sociopath. This image has been heightened not only by the media, but by the some of the actions of the virus writers themselves. Public communications from the writers, in the form of echo-mail messages, often seem to indicate they are intent on doing as much damage as humanly possible. Their electronic publications have in the past reinforced this, and the very fact that they release viruses may seem to confirm it: these people are bad. This paper argues that this is a gross oversimplification of the situation, and that the virus writing aspect of these individuals is not sufficient to characterize them into one group simply labelled 'unethical people'.

We will show that virus writers are not all the same as each other as far as their stages of

ethical and moral development; we will show that some virus writers are within normal ethical developmental model ranges as defined by Lawrence Kohlberg's model of moral development [2].

Return to Top

The Generic Virus Writer

Stereotyping is pervasive. It is especially prevalent when a new kind of entity emerges, or a new kind of person. As there is little reliable information about such new kinds of people, differentiating between them is difficult. Thus, there is a tendency to assume not only that there is some stereotype, but also that anyone who can be classified as belonging to the newly perceived group is to all intents and purposes like all the other members of that group. This often happens when the newly emergent group is primarily composed of young people.

Moreover, such stereotyping is often accompanied by generalised value judgements. In the case of virus writers, a common assumption is that they are all bad. While it is certainly true that the distribution of malicious software is a bad act, and that many virus writers are motivated by bad or even criminal intentions and desires, it is dangerous to assume that this is true of every person who ever writes a program that can be classified as a virus. The problem of dealing with the danger posed by the distribution of malicious software is not simplified by failing to recognise that the people who write viruses do not form a homogeneous group. They are a diverse group. If we are to address the problem, we must first recognise its true nature. We must discover how different virus writers operate, and how they are as people. To this end, we will examine similarities and differences of four individuals involved in the virus writing culture.

Return to Top

Ethical models

Ethics is sometimes promoted as one solution to the problem of people writing viruses. To explore what part ethics may play in virus writing, we will examine four virus writers using a model of ethical/moral development as a base for comparison. We chose to use a model of ethical development that was universal and longitudinal. Virus writers come from diverse cultures, so the use of a universal model is desirable. We chose Kohlberg's model for its universal characteristics. The research done by Kohlberg was not only cross-cultural but longitudinal; it was performed over a time period of 12 years. Based on this research, he designed a six-step ethical classification model, which shows a fixed sequence of changing responses with increasing age. It has been shown to be based on experimental and longitudinal evidence, and is based on 'methods of thinking' rather than

individual actions or decisions [3].

Return to Top

Kohlberg's model

Kohlberg's ethical model provides age trends in three moral levels of development, with two stages within each level. These levels/stages of development are defined as:

Level 1: Pre-conventional morality. At this level, morals are external.

- Stage 1

  The first stage consists of the punishment and obedience orientation (i.e. there are no real rules; the seriousness ofa 'bad' act depends on the consequence of the act). This stage is sometimes referred to as the punishment orientation stage. 'Right' is being obedient to power and avoiding punishment at all costs.

- Stage 2

  In stage two, instrumental orientation surfaces (being good to get a reward or satisfy a need). In Kohlberg's study, 80 percent of moral judgements of ten year-olds are in this category. This stage is sometimes called the naive reward orientation stage. 'Right' behaviours include taking responsibility for oneself, and letting others take responsibility for themselves.

Level 2: Conventional morality. Parents, social groups and peers play a large role of influence at this level. Being 'good' is important. Rules may appear 'internalized', but they may be internalized to avoid punishment or to gain the approval of others.

- Stage 3

  In the third stage of development, actions are judged on the merit of their intent. A person has to be able to recognize the point of view of others to progress into this stage. This stage can be referred to as the good-boy/good-girl orientation. 'Right' is having a right motive, and a concern for others.

- Stage 4

> In the fourth stage, one 'accepts authority', not only because of the possibility of punishment, but out of a sense of duty to obey rules and maintain social order. This stage represents authority orientation. The rules of a society are important in this state: 'Right' is keeping the rules of the society.

Level 3: Post-conventional morality. Self-accepted moral principles are the mark of this level. In stage five and six, morals are internalized. The stages in level three involve development of personal codes of ethics.

- Stage 5

> Judgements become more flexible in stage five. Rules must be impartial, and 'The Welfare of the Many' becomes paramount. This stage is sometimes referred to as the social-contract orientation stage. 'Right' is keeping the contract.

- Stage 6

> In stage six, the individual defines right and wrong on the basis of his/her own ethical principles. Normative ethics, based on self-chosen principles are applied in all situations. This form of development is consistent with the ability to perform formal operations (the highest level of cognitive development) [4]. This stage represents the morality of conscience. 'Right' is an obligation to the universal principles of equality, justice and respect for persons.

> [Stage 6 may be viewed as a hypothetical construct as no group seems **consistently** able to fit in this slot; in fact, this state is often eliminated from some versions of the model. It is however, a desirable stage, and it is possible for some people to function at this level some of the time.]

Return to Top

Gender Issues

While Kohlberg's model is well suited for the purposes of this study, the Gilligan model can be helpful in addressing gender issues of virus writing from the standpoint of ethical

development.

In conversations with dozens of individuals involved in the virus writing culture, we have found only two instances of 'direct' female involvement. One was the girlfriend of a virus writer, and one was a woman who was involved with the virus writing group NuKE. However, it is uncertain as to whether or not she ever produced any viruses. According to Gilligan, females progress through different states of moral reasoning. 'Females are socialized to equate 'goodness' with self-sacrifice more than are males' [1]. Gilligan's three stages of moral development are described in the next section.

Return to Top

Gilligan's model

- Stage 1

  Self-interest. At this stage the needs of others are ignored.

- Stage 2

  Self-sacrifice. At this stage, women sacrifice their own needs/desires for the well-being of others.

- Stage 3

  Non-violence, mature thinking; compassion and universal good.

Gilligan states that while male and female children go through stages of being subject to parental authority and then peer pressure (where right and wrong are determined by the groups they belong to), females do not progress through the utilitarian and deontological stages. Instead, they view moral decisions in terms of human interdependency and needs as well as justice and rights. Females involved in the virus writing culture are typically treated as inferior by a disproportionate number of members of the culture. Sexual slurs and harassment are common. Women in this culture do not appear to be able to pursue their goals independently of men. There appears to be little attention to concepts of equality, or even a pseudo-equality.

While there are opponents to her theory [5, 6, 7], we propose it would help partially explain the marked absence of female virus writers.

Return to Top

Methodology

While we had access to a varied population of virus writers, and the opportunity to draw a sample from the population, the measurement of the sample proved to be extremely complex. Rather than use computed descriptive statistics to make only inferences about the similarities within the population, we also chose to examine the differences by using case studies.

We have adopted an inductive approach so that we can learn who the 'generic virus writer' might be by observing instances of actual virus writers. We believe this is a more sound approach than trying to produce a characteristic profile to which actual writers can later be matched. We wished to avoid making many assumptions about what might or might not be in such a characteristic profile until we had examined some real cases.

The virus writing community is relatively small in comparison with other underground communities such as the hacking and phreaking communities. There is no way to define the population exactly; however, if we consider viruses that are known to exist, we can estimate there could be at most 4500 virus writers, if one person wrote each virus. We know that more than a few of the viruses are written by the same person. For instance, there are a number of viruses that are known to have been written by someone calling himself Dark Avenger; so, not each of the viruses we know may have an individual author.

When we look at the viruses 'in the wild' as opposed to research viruses or viruses which are only sent to product development companies for inclusion in virus scanners, we find approximately 150 examples. Of those, if we estimate 100 as by different individuals, the responses we gathered would constitute response by approximately half the writers of viruses 'in the wild'. Of course, we have no way of knowing exactly who wrote what, or if all of our respondents actually did write the viruses they claim. It is quite possible that there were respondents who merely wished to participate, or who in fact deliberately wished to bias or discredit this study. However, we do know that of our four case studies, every one of them has authored viruses that have appeared in the wild.

We distributed the survey directly to 47 virus writers known to us. From those 47, we received 18 individual responses to the survey, which was distributed to underground bulletin boards in the United States, Germany, Australia, Switzerland, Holland, and South America. In addition to the 18 responses we received to the survey directly, we talked to an additional 43 individuals involved in the virus writing culture who did not wish to complete the surveys, but who consented to talk about their motivations and histories. We received 3 negative (hostile) responses. Total responses: 64.

The confidential survey (Appendix 1) was comprised of questions including requests for information on social interactions with peers, relationships with parents and other authority figures, as well as exercises in cognitive reasoning. We were concemed

primarily with the methods of thinking used as opposed to the 'right' answers. The actual answers were not as important as the reasons given for the answers. Other questions concerned age, employment and educational history. Questions were asked to provide us with data regarding the respondents relationships with parents and peers. The response to these questions enabled us to see how the individual considers himself to 'fit in' in both his immediate society and society in general. We also asked questions about conflict resolution to enable us to see what processes the individual uses to solve problems involving other people.

In order to illustrate reasoning abilities, the following questions were asked:

1. You have four coloured placoloured plates: Red, Blue, Yellow, ae, ease tell me all possible color combinations.
2. What number is 30 less than 3 times itself. When you answer this, please write (or type) for me each step of reasoning you used to arrive at your answer.

The responses to the these types of questions provide a window into the reasoning abilities of the individual. Reasoning abilities have been shown to related to moral development [4]. We asked the respondents to tell us not only the 'answer' but to describe for us how they obtained the answer.

We included the classic scenario used by Kohlberg when studying the ethical development of individuals:

> Read and consider carefully the following scenario.
>
> In Europe, a woman was near death from a special kind of cancer. There was one drug that the doctors thought might save her. It was a form of radium that a pharmacist in the same town had recently discovered. The drug was expensive to make, but the pharmacist was charging $2000, or 10 times the cost of the drug, for a small possibly life-saving) dose. Heinz, the sick woman's husband, borrowed all the money he could, about $1000, or half of what he needed. He told the pharmacist that his wife was dying and asked him to sell the drug cheaper, or to let him pay later. The pharmacist replied, 'No, I discovered the drug and I'm going to make money from it.' Heinz then became desperate and broke into the store to steal the drug for his wife.
>
> Should Heinz have done that?
>
> Now that you have read it, and considered it, please resolve the moral dilemma. That is, what are the problems in the story? What problems does each person have to deal with? Who is wrong, right, and why?
>
> When you write your response, please include the following points:
>
> Should Heinz be punished for stealing the drug? Did the pharmacist have

the right to charge so much? Would it be proper to charge the pharmacist with murder? If so, should his punishment be greater if the woman who died was an important person? What would you have done if you were Heinz?

We intended the questionnaire to provide information directly as well as indirectly, as we did not want to make too many initial assumptions.

We received very detailed responses to the questions. For example, to our question 'Which number is 30 less than 3 times itself?' we received detailed accounts of the process by which the conclusion was derived. One respondent stated he arrived at this answer by substituting one number after another until one worked. Another respondent provided us with an algebraic equation.

```
x = the number in question;
x = 3x - 30
0 = 2x -30
-2x = -30
x = 15

So the answer is 15.

Proof:
15 x 3 = 45
45 - 30 = 15
15 = 15
(reflexive property I think)
```

The differences in the responses illustrate the difference in the cognitive reasoning abilities of the individuals which in turn correlate to the level of moral development as proven by Kohlberg. According to further research by Carol Tomlinson-Keasey and Charles Keasey [8] and Deanna Kuhn [9], individuals who demonstrate at least some formal operational skills on cognitive tests have necessary skills for development of postconditional morality.

To develop the case studies, we exchanged electronic mail with some of the respondents following collection of the survey data. These interviews used both structured and unstructured formats. We talked with some respondents electronically using Internet Relay Chat, and the UNIX 'talk' command. Some of the respondents telephoned us directly. We conducted interviews with some subjects in person. In some cases, where the identity of the subject was totally unknown and he did not wish to be identified via mail or talk sessions where we could netstat him, we arranged for him to login to IRC via an anonymous host. We then talked on IRC in a private channel.

These interviews provided us a more detailed insight into the life history of the

individuals who had consented to be case studies.

Return to Top

Virus Writers

We will attempt to provide a broad classification of virus writers according to a number of parameters. Our intention is not merely to provide an abstract schema of how such a group might be differentiated, but to see how actual virus writers may differ. In particular, we are interested in trying to establish how virus writers develop and progress from early beginnings to whatever it is they end up doing. To this end, we will examine four cases studies conducted recently. These case studies are all of people who have at some time written a virus. However, as will become apparent, each of these people is very different from the others. By examining these differences, we hope to shed some light on the notion of the 'generic' virus writer, and to ask whether or not such a concept is valid or useful.

The four initial categories we chose can be described as follows:

- The Adolescent

  Virus writer aged 13-17; has written at least one computer virus; has distributed at least one computer virus into the wild.

- The College Student

  Virus writer aged 18-24; has written at least one computer virus; has distributed at least one computer virus into the wild. Student in university or university level classes.

- The Adult/Professionally Employed

  Post-college or adult, professionally employed; has written at least one virus; has distributed at least one virus into the wild.

- The Ex-Virus Writer

  Virus writer who has written and distributed one or more computer viruses. The viruses must have been found in the wild; the author must have supplied sufficient proof to enable determination that he did indeed write the virus;

> there must be no evidence that he has written or continued to write viruses for a period of at least 6 months prior to commencement of this research.

The individuals who were chosen as case studies were taken from the selection of virus writers in their respective groups. We note that in each group, while the ethnographic data varies, the responses to questions related to ethical development and cognitive reasoning remained constant between the individuals we selected and the others in their group.

Return to Top

The Adolescent

The case study selected is a 16 year-old unemployed male high school student. He states he is one of three children, and lives with both parents in what is considered an upper-middle-class home. He describes his relationships with his friends as daily interactions. He does not express an interest in sports. He has no formal ethical education. He states his friends are very self-contradictory, and that they argue frequently. The arguments appear to be of a philosophical nature; what is worthwhile, what is valid, what is reasonable. He displays a strong conviction against racism, and bias. He describes his friends as having no morals. He states he does not play computer games other than a game that came with Windows. His responses to methods of conflict resolution are unclear. His response to ethical reasoning dilemmas fall in the range of stage 2, instrumental orientation/hedonism. For instance, one of his responses to whether or not it was OK for Heinz to steal the drug was 'Yes. It was for a good cause'. He states that destructive code is unethical, and that he has never researched a virus by his own definition of 'research'. He still writes viruses, and his viruses have been found in the wild. When asked how he felt regarding his viruses that have been in the wild, he responded:

> Generally, I feel almost sorry for the people who are infected with my viruses. I believe only three or four of my twenty some odd viruses have been found in the wild. The rest were distributed via underground bulletin board systems.
>
> One of the viruses, xxxxx,xxx (named by F-Prot), was found on a CD-ROM entitled (name deleted). I'm not exactly sure how it got there, but I know for certain it originated on Canada Remote Systems On-line located in Toronto. The bait file was probably uploaded to that bulletin board by a local virus enthusiast.

Conversations with this individual indicated that he has a respect for his parents and for authority to some degree. He demonstrates in his communications a knowledge of what is

right and what is wrong, and expresses that things that are illegal are wrong. He indicates that he does not favour destructive viruses, yet seems to not have any problem with his own position of having released viruses into the wild. He is respectful to other people, and tends to be a leader in group situations.

His responses and electronic communication were at all times very polite, respectful and thoughtful.

Return to Top

The College Student

The case study selected is that of an 18 year-old virus writer. The subject is unemployed and living on his own. He grew up with one sister in a moderately well-to-do family. He enjoys martial arts and has practised them for several years. He describes his relationships with his friends as close, and open. He states his relationships with women are good, and that he spends time daily with his girlfriend. His relationship with his parents is described as very good, with the normal disagreements. Conflict resolution on the part of this person is conciliatory and mature. He states that he values the diversities that his friends possess. When asked about the influence of others on his life, he responded, 'In virus writing, I respect such authors as Dark Angel and Masud Kafir not only for their technical programming skills, but also for the fact that their major viruses are not destructive'. He indicates that while he recognizes using pirated software is not right, he occasionally uses pirated software: he buys software when he can afford it. While he used to play computer games, he claims he now no longer has time.

His ethical background consists of study of Kant, Mill and Aristotle. He states he feels he is most like Mill, in that one should be able to have as much freedom as possible without harming another. He states he knows he fails at this sometimes. His responses to ethical dilemma questions were at level 4, which would place him at slightly higher than average position according to Kohlberg's model.

> I feel that yes, Heinz should steal the drug as it will save his wife (this would be my first priority) if there is no other way to get it, he is in the wrong legally and should be punished if caught.

He defines virus research as a search for truth/facts, objective series of tests. He states some 'researchers' are actually merely collectors who sell their viruses for profit, monetary or otherwise. Where and to whom the viruses go is named as one ethical issue. The possibility of release, as well as destruction/use appears as another issue. He cites money for viruses and/or anti-viral software as a grey area.

He states he began writing viruses at the age of approximately 15 when he found the Stoned virus. He became competent at assembler and has written viruses in the past three

years. He writes viruses for text publication as well.

Return to Top

The Adult (1)

[(1) Adult males are typically at stage 4 and sometimes 5 [10, 2]. The adultssurveyed/observed did not demonstrate five or six at any time of ethical development, unlike some of Kohlberg's subjects.]

The adult case study is a single male, who describes himself as living with a ladyfriend. His income is listed in the middle-income range; he is professionally employed. He is one of four children, and has completed high school, with some college. He states the majority of his friends are female. He describes his relationship with his parents as very good. His relationships with friends are described as social interactions of a casual nature. Conflict resolution is addressed in terms of power issues. He indicates hypocrisy and unethical actions as stimuli for provoking him to anger. For instance:

> District Attorney crusades against pornography at election time, has bookstore operator or adult BBS operator arrested, confiscates/destroys merchandise/money/equipment but does not pursue the case. Gets re-elected somehow.

He states his friends do not care much about morals. He states he plays computer games perhaps 4-5 hours per week, if that much.

He states he does not use pirated software. The responses to cognitive reasoning questions, and to questions regarding ethical dilemmas place him at stage four, where obligation to law is above special interests. He describes virus writing as a pointless exercise. It is not certain whether he has continued to write viruses, although he has stated he does not really enjoy programming. He stated he thought programming would get him a good job, which it did not. This individual is involved in virus distribution, which he states is 'not illegal'.

Return to Top

The Ex-virus Writer

The ex-virus writer is a college student; the only child of an upper class family, raised in an atmosphere where academic performance was greatly valued. He has never been formally employed, but has worked as a volunteer at a library (shelving books), and as a volunteer at a hospital where his job was to help handicapped/geriatric patients. He states he was active in track, and describes his relationship with his girlfriend as good.

However, he states he did not have a girlfriend until his last year of high school, as he was by his own definition, 'shy'. His narration of his peer relationships and interactions closely mirror those of the teen virus writer; he states his friends do not have morals that are very developed for the most part: '..most of my friends have not had a reason to question the morals they have been brought up with, so they have not fully examined their morals. Then again, neither have I, although I am trying to do so now'. His relationship with his parents is described as 'not good'. He described them as controlling individuals who were performance-motivated.

He addresses conflict resolution logically; problems are identified, then solved. He does not tolerate hypocrisy. Throughout our conversation, which was conducted in person, he frequently questioned his own morals and values. He stated that he did not 'think about it' (the morality of releasing or writing viruses) when he was actually doing it. I asked him specifically if his viruses were destructive. He stated 'They can't be!'. Like the teen and college student profiled earlier, he expressed a marked dislike for destructive code. He began writing viruses out of curiosity. He stated he quit because he did not have any time for it. He states he sees himself as somewhat 'obsessive', although his virus writing did not take a lot of his time. He states he does not use copyrighted software and does not play computer games any more (he used to play them but they became too big to run on his computer). He defines research as follows:

> Doing significant work towards meaningful results in a field. Running scanners is not research. Compiling test results is not research. Disassembling viruses is not research. Writing a new scanner is not research. Examining the behaviour of viruses and their consequences is research. Developing and implementing new techniques of detection and cleaning is research. Classifying viruses in a reasonable fashion is not research, but it is meaningful science.

He states he cannot say virus writing is ethical, nor can he state it is unethical, as

> there is some degree of that (lack of ethics), but I usual don't think of it as an ethical issue. I recognize that there is a degree of irresponsibility associated with most virus writing.

He gave the following reason for deciding to stop writing viruses:

> I decided to stop primarily because I no longer have the time to write. My productivity in writing viruses was directly proportional to my level of boredom. I contend that my real-world impact is low. None of my viruses are common in the wild and I have given nobody any information that they couldn't have figured out on their own. My philosophy has always led me to create viruses designed to be non-destructive and I don't intend for anyone to be hassled with one of my viruses. It's a hobby, and I just don't

> have time for it anymore. I've also gotten bored with viruses; they're interesting for a while, but then there isn't much more to do with them. I really don't know what significantly more interesting stuff there is to do with viruses.

He made the following suggestion for stopping viruses from being written/distributed:

> Demystify them. If you want people to stop, demystify them. All that will be left then are malicious people, and you can deal with them.

He stated he quit because he simply had too many other things to do. He also indicated that he did not want to carry the 'stigma' of writing viruses, and that had he realised earlier (the consequences), he would have been smarter. His feeling was that people could be discouraged by demystification and 'character'. He stated that responsible computing should be taught very early.

He states respect for others is important.

> People who cut me off on the road used to undergo a thorough drubbing: bright lights, following, later cutting off and trapping. This was before I realised how dangerous a game it was that I was playing.

He states he is angered by boasting that has no foundation.

> Rock Steady is an example. I wrote an expose file on him and all his code that I was considering giving out, in which I trashed all his code and traced its origins... people should not get respect by others if they have nothing to back it up with.

I had approximately 4.5 hours of interview with this individual in the naturalistic setting, as well as many hours of electronic interchange and telephone conversations. I was impressed with his genuine openness, intelligence, and his apparent honesty and thoughtfulness. His response to the survey was 13 pages of text, which we discussed at length.

Using the Kohlberg model, his ethical/moral development appears to be at stages 4, and 5 - occasionally 6, in both thought and action. This is slightly deviant as he is not at the age where males normally would exhibit these levels/stages. However, his responses clearly place him there and we have no reason to doubt them.

He states for instance that the best reason to observe a speed limit is to prevent yourself from losing control of the car. His responses to the Heinz dilemma question were:

> Heinz clearly should not have stolen the drug, even though it meant his

> wife's life. However, this is based upon our society assumptions of legality and does not reflect my own moral view... The pharmacist has a right to charge a high price, but he should be morally obligated to charge an affordable rate... Heinz should certainly be punished for stealing the drug. Stealing, after all, is still stealing and it is still a crime. There can't be any 'exceptions' to the law for such cases; otherwise, what would distinguish 'good' stealing from 'bad' stealing? And would people think they're doing 'good' stealing and get punished? However, the sentence should be lenient to reflect the circumstances.

What do these case studies tell us? We see that the individuals are different in personal characteristics. We see that the adolescent and college student are at developmental levels that would be expected for their age. We see the ex-virus writer at the stage (or above) one would expect someone with a mature view to have, slightly above the norm for his age. We see the adult at an ethical/moral development stage below what Kohlberg's model states is the norm.

For purposes of comparison, we solicited control subjects who never wrote viruses. They were also different in personal characteristics, and their ethical development according to Kohlberg's model was consistent. However, the adult control subjects placed in the category defined by Kohlberg as normal for their age, unlike our virus writing subject. This does not enable us to conclude anything, but is worth further study, to see if there is indeed any connection. At this time, all we have proven is that not all virus writers are the same, and that some virus writers are normal as far as ethical development goes for their ages.

While these individual case studies would indicate some of the individuals had some evidence of a relatively high ethical developmental stage, this does not tell us how they will actually act in a given situation. Ethical judgements are normative in nature. Of course, in real life, we often make different decisions than we do in theory [11, 12, 13, 14, 15]. This explains why an individual could think it is 'wrong' to write computer viruses, and yet write them and still have ethical standards which generally appear to be normal or above normal for their age groups. According to research done by Lawrence Walker and his team of researchers, even when people do operate at different levels on hypothetical/real life dilemmas, they use reasoning at adjacent stages on the types of issues [16]. The responses we received agree with Walker's work.

Research performed by Hugh Hartshorne and Mark May [17] provides an investigation of the moral character of children aged 8-16 in a variety of settings. This study also showed that the behaviour of a person in one situation did not predict his/her willingness to conduct the same behaviour in another situation. Later research performed by Nelson, Grinder and Mutterer [18] and Roger Burton [19] found that the aspects of morality do indeed become more consistent as age level increases.

What sorts of interactions and social experiences allow a person to progress to the more mature levels of ethical development where their actions are more conciliatory with their beliefs and values? In Kohlberg's study, we see that transitive interactions consistently result in change [4]. These interactions, which are social experiences, facilitate moral growth by introducing cognitive challenge. These social and verbal exchanges require performance of mental operations on the reasoning abilities of ones peers. We can observe this form of interaction in the descriptions our college student gave concerning his interaction with his peers. We see further evidence of this progression when we review the sort of interactions described by the ex-virus writer. This sort of exchange is necessary for progression to the higher levels of ethical reasoning. At a higher level of ethical development, individuals' ethical values and actions begin to come closer together. While some don't ever get there, most do. Some even progress to higher stages, such as stages 5 and 6.

Further studies conducted by Kohlberg and his associates have shown that the majority of non-criminals are classified in stages three and four, while a majority of criminals are classified in stages one and two [20]. People who obey law to avoid punishment or who are primarily motivated by self interest appear more likely to commit crimes than those who see the law as beneficial to all of society. Research efforts on youth have shown that a significant number of deviant youth were in categories one and two, while non delinquents rank higher [21].

Return to Top

Conclusion

Based on this research, which is by no means definitive, we have observed that virus writers are not a homogeneous group. They have characteristics similar to many populations. They vary in age, income level, location, social/peer interaction, educational level, likes, dislikes and manner of communication. The ethical developmental models of the young adolescent and college age virus writers are within the norms for the age groups of the individuals. From the data collected, it is uncertain what predisposes them to writing and releasing computer viruses. There is only one common characteristic, and that is that their ethical development appears to be within established norms. This is not the case with the adult participant in the culture. Where adults in the control group exhibit level 3 stage 5 of ethical development, not one of the adult virus writing respondents answered any of the questions in a way that would lead us to believe he/she regularly functions at level 5 development. What does this mean? There are other segments of the population that do not function at this level, and they are not judged to be ethically 'deficient'; however, this departure from the norm would seem to be one factor worth further consideration. We can conclude that there is no homogeneous group to which 'The Virus Writer' conforms. There are too many observable differences to categorize them into a generic construct. However, we can learn from the observations.

In our study, different manners of thinking were observed; different motivations were observed. No one seemed to target government or military as the 'subject' of their viruses. In fact, with the exception of anti-virus product developers, there was no direct 'targeting' mentioned or implied in any of the interactions. 'The Enemy' was virtually non-existent to the teen and college student virus writers. 'The Enemy' to the adult respondents consistently appeared to be 'Society'. The three ex-virus writers varied in their perception of 'The Enemy'. One saw the enemy as society, but seemed to feel that he could not 'win' this battle; one stated there was never an enemy and the third stated that the enemy was 'within' the individual.

Female participation in the virus writing culture appears virtually non-existent. It is possible that female participation may increase, following patterns similar to female involvement in other forms of youth deviant behavioural models.

There are a number of social issues which are related to what is often perceived as the isolated act of 'computer virus writing' (used here to mean, distribution to unwilling/unknowing persons). Environmental and social issues including abuse of substances, child abuse, education, etc., are factors to be considered when assessing any juvenile crime or dysfunctional behaviours. Because of this, the multi-disciplinary or interdisciplinary study of this phenomenon would appear to be the one that will yield the most effective conclusion.

There are some similarities between the disfunctional behaviour of distribution of computer viruses to unknowing/unwilling persons and forms of juvenile delinquency. And, as with the social phenomenon of delinquency, we do not know why some persons involved in this subculture become chronic 'career' offenders, beginning early and continuing into adulthood. We do not know what factors contribute to the continuation of the activity, or what factors can contribute positively to the desistance or termination of the activity. One theory that is often advanced is the theory of ageing out, or spontaneous remission. In work by Michael Gottfredson and Travis Hirschi, it is proposed that age-crime relationships are constants: not only do chronic juvenile offenders commit less crime as they get older, but all persons commit less crime as they age. Therefore, age/crime correlations are irrelevant to the study of crime [22, 23]. Of course, there are opposing views which purport that the earlier a person demonstrates antisocial tendencies, the longer they will continue to commit these acts. This sort of longitudinal theory deals with life-cycle of delinquency/anti-social behaviour, and attempts to correlate age/crime. Deterrence theory proposes that the choices young people make can be controlled by threat of punishment: the more severe, certain and swift the punishment, the more the deterrence value. Proponents of such theory support laws to impose severe penalties on virus writers. However, it is not certain that such strategies work, and in fact they may be counterproductive. According to research published in the Journal of Criminal Law and Criminology,

> Little reason exists to believe that crime and delinquency can be

eliminated merely by the fear of legal punishment alone. More evidence exists that fear of social disapproval and informal penalties, criticisms, and punishments from parents and friends may actually be a greater deterrent to crime than legal punishments[24].

Sociologist Jack Katz feels the seduction of crime is a prime motivation for anti-social acts [25]. Research conducted in Toronto, Canada by John Hagan and Bill McCarthy supports this theory, which places at least part of the cause for this behaviour on situational inducements [26]. Cultural deviance theory maintains that certain actions are performed because the individuals adhere to the value system within their own subculture. We can consider dealing with the persons who distribute viruses maliciously in the same ways as we deal with others who do what we perceive to be malicious acts. This includes clarifying our own positions on what constitutes malicious action; constraint, degree, intent, knowledge, 'bad tendency' and clear and present danger.

Return to Top

Bibliography

1. Carole Gilligan, 'In a different voice: Psychological theory and women's development', Harvard University Press, 1982.
2. Lawrence Kohlberg, 'State and sequence: the cognitive-developmental approach to socialization', in D. A. Goslin (Ed), Handbook of Socialization Theory and Research, Chicago: Rand McNally, 1969.
3. A. Colby, L. Kohlberg, A. Gibbs & M. Liberman, 'Monographs of the Society for Research in Child Development, 48', (1 & 2 Serial No. 200), 1983.
4. David R. Shaffer, 'Developmental Psychology - Childhood and Adolescence'. Brooks/Cole Publishing Company, 1989.
5. S.J. Thoma, 'Estimating Gender Differences in the Comprehension and Preference of Moral Issues', Developmental Review, 6, pp 165-180, 1986.
6. L. J. Walker, 'Sex Differences in the Development of Moral Reasoning: A Critical Review', Child Development, 55, pp 677-691, 1984.
7. L. J. Walker, 'Sex Differences in the Development of Moral Reasoning: A Rejoinder to Baumrind'. Child Development, 57, pp 522-526, 1986.
8. Keasey, C. B. & Keasey, C. 74. Carol Tomlinson-Keasey and Charles Keasey. 'The Mediating Role of Cognitive Development in

Moral Judgement', Child Development, 45, pp 291-298, 1974.

9. Deanna Kuhn, Lawrence Kohlberg, Langer, and N. Haan, 'The Development of Formal Operations in Logical and Moral Judgement', Genetic Psychology Monographs, 95, pp 97-98, 1974.
10. C. Holstein, 'Irreversible, Stepwise Sequence in the Development of Moral Judgement: a Longitudinal Study of Males and Females', Child Development, 47, pp 51-61, 1976.
11. A. Blasi, 'Bridging Moral Congnition and Moral Action: A Critical Review of Literature', Psychological Bulletin, 88, pp 1-45, 1980.
12. Lawrence Kohlberg, 'Moral Stages and Moralization: Cognitive-Developmental Approach to Socialization', in T Lickona (Ed), Moral Development and Behaviour: Theory, Research and Social Issues, New York: Holt, Rinehart and Winston, 1975.
13. J. W. Santrock 'Moral Structure: The Interrelations of Moral Behaviour, Moral Judgement, and Moral Affect', Journal of Genetic Psychology, 127, pp 201-213, 1975.
14. E. A. Nelson, R. E. Grinder and A. M. Biaggio. 'Relationships Between Behaviour, Cognitive Developmental, and Self-Report Measures of Morality and Personality', Multivariate Behavioural Research 4, pp 483-500, 1969.
15. I. J. Toner and R. Potts. 'Effect of Modelled Rationales on Moral Behaviour, Moral Choice and Level of Moral Judgement in Children', Journal of Psychology, 107, pp 153-162, 1981.
16. L. J. Walker, B. de Vries and S. D. Trevethan. 'Moral Stages and Moral Orientations in Real Life and Hypothetical Dilemmas', Child Development, 58, pp 842-858, 1987.
17. H. Hartshorne & M. May, 'Studies in the Nature of Character', volume 1: 'Studies in Deceit' volume 2: 'Studies in Self Control' volume 3: 'Studies in the Organization of Character', New York, Macmillan, 1928-1930.
18. E. A. Nelson, R. E. Grinder and M. L. Mutterer, 'Sources of Variance in Behaviour Measures of Honesty in Temptation Situations: Methodological Analyses', Developmental Psychology, 21, pp 265-279, 1969.
19. R. V. Burton, 'The Generality of Honesty Reconsidered', Psychological Review, 70, pp 481-499, 1963, 1980.
20. Lawrence Kohlberg, K. Kauffman, P. Scharf and J. Hickey, 'The Just Community Approach in Corrections: A Manual', Niantic Conneticut, Conneticut Department of Corrections, 1973.

21. Scott Henggeler, 'Delinquency in Adolescence', Newbury Park, California, Sage Publishing Company, 1989.
22. Michael Gottfredson and Travis Hirschi, 'The True Value of Lambda Would Appear to Be Zero: an Essay on Career Criminals, Criminal Careers, Selective Incapacitation, Cohort Studies and Related Topics', Criminology 24: pp 213-34, 1986.
23. Michael Gottfredson & Travis Hirschi, in Lawrence Cohen and Kenneth Land, 'Age Structure and Crime', American Sociological Review 52, pp 170-183, 1987.
24. Van den Haag
25. Jack Katz, 'Seductions of Crime', New York, Basic Books, 1988.
26. Bill McCarthy and John Hagan, 'Mean Streets: The Theoretical Significance of Situational Delinquency among Homeless Youths', American Journal of Sociology 3, pp 597-627, 1992.

Rctum to Top

About the Author

Sarah Gordon's work in various areas of IT Security can be found profiled in various publications including the New York Times, Computer Security Journal and Virus Bulletin. She is a frequent speaker at such diverse conferences as those sponsored by NS A/N1ST/NCSC and DEFCON. Recently appointed to theWildlist Board of Directors, she is actively involved in the development of anti-virus software test criteria and methods. She may be reached as sgordon@low-levcl.format.com

Return to Top

back to index

ISSN 0956-9979 SEPTEMBER 1995

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **All the colours of the Rainbow.** A new virus, Rainbow, has appeared which utilizes circular extended partitions. What does this mean for the user? See the analysis on p.12, and our tutorial on the subject on p.14.

• **Genus and species.** A hoary problem for anti-virus researchers has always been the issue of virus naming. Great efforts are being made to standardise this process, and the first section of a two-part article by Dr David Hull (p.15) clarifies what is involved.

• **Detecting a new way.** *Cheyenne Software* is exploring pastures new; their latest product is *InocuLAN for Windows NT*. How does this product compare with the others in this growing field? Turn to p.18 to find out.

CONTENTS

EDITORIAL
When Techniques Jump Fences 2

VIRUS PREVALENCE TABLE 3

NEWS
1. *C+P+N+A+V = ?* 3
2. *ESaSS* and *Reflex* Announce Alliance 3
3. *VB '95*: Boston on the Horizon 3

IBM PC VIRUSES (UPDATE) 4

INSIGHT
Igor Grebert: Carpe Diem 6

VIRUS ANALYSES
1. What a (Winword.)Concept 8
2. Byway: The Return of Dir_II 10
3. Rainbow: To Envy or to Hate 12

TUTORIAL
Circular Extended Partitions: Round and Round with DOS 14

FEATURE
Computer Viruses: Naming and Classification 15

PRODUCT REVIEWS
1. *InocuLAN for NT* 18
2. *IBM AntiVirus* 21

END NOTES & NEWS 24

EDITORIAL

When Techniques Jump Fences

This month's *Virus Bulletin* is perhaps not its usual self. Outwardly it appears the same, but inside, things are different, for it documents not one, but *two* new attack techniques which have appeared in recent weeks and months (see p.8 for an analysis of Winword.Concept, and pp.12-14 for information on the Rainbow virus).

This situation is somewhat analogous to the famous truism of waiting two hours for a bus, and then having three come along at once. New techniques are few and far between, but, like buses, they travel in packs.

"new techniques are few and far between, but, like buses, they travel in packs"

A fairly good working definition of the expression 'new technique' is one which forces anti-virus manufacturers to make some design change to their products. A new polymorphic file infector does not, these days, meet this criterion - the vast majority are very similar, contain nothing new, and (once the producers have updated the virus databases of their products) present no great problem.

Both Winword.Concept and Rainbow meet this criterion, and so will (or should!) provoke some thought from anti-virus producers. Winword.Concept may induce concerns about whether or not to scan *Microsoft Word* files (.DOC and .DOT) - this in itself introduces a world of problems, as the formats of such files are non-obvious. However, Rainbow, which prevents a clean boot, appears to be the more awkward of the two.

The concept of clean booting before attempting to remove viruses is so fundamental to the way the current systems work that a virus which consistently prevents it reliably is bound to cause problems. Rainbow does this on those versions of DOS which are most 'in the wild' (at least in the Western World) - *MS-DOS v5* and above. It is quite within the realms of possibility that a site infected with such a virus would not have clean boot disks of a version earlier than that.

There is a world of difference between an anti-virus product stating that you must have a clean boot disk in order to clean up any infection, and that same product stating that you must have a variety of clean boot disks containing different versions of DOS to suit every occasion. The former is widely accepted, because this is how the system works - there is no real need for a product to deactivate a virus in memory, as a clean boot has always been the simpler course. Although the latter is much more annoying, it is possible that it will be the way people have to move.

In this, as much as in anything else, it is true to say that there is very little which is truly new. The concept of circular partition sectors (*à la* Rainbow) had already been described by the early 1990s, and the idea of a macro virus had been described (albeit in relation to *Lotus 1-2-3*) even before that. However, these techniques have now crossed the barrier dividing the world of research speculation from that of real viruses.

It is interesting to note how long such a crossing has taken - the ideas have been knocked around for so long, and yet have taken this many years to reach the other side of the fence. Well, yes and no: the theories have no doubt been known amongst the virus writers for almost exactly the same length of time as the researchers have known about them.

Whether or not these particular techniques become prevalent in the wild (either by way of the viruses described here, or by other viruses, developed later, which use the same ideas) remains to be seen. However, it does seem highly probable that more viruses using these techniques *will* appear, and this will only serve to highlight the need for anti-virus developers to find ways to make their products deal with them.

One thing is certain - jumping up and down and panicking about the end of the computing world as we know it is not going to help. Neither of these viruses, or their techniques spell doom for the anti-virus industry or modern computing; they simply mean we may have to think about some things slightly differently from now on.

NEWS

C+P+N+A+V = ?

Speculation on the future of *Central Point Anti-Virus* has risen once again, with the imminent release of *Microsoft's Windows 95*. *Central Point Software* was subsumed by the giant conglomerate *Symantec Corporation* last year, and ever since then, industry has been discussing whether or not *CPAV* would be incorporated into the current *Symantec* product, *Norton Anti-Virus* (*NAV*).

Fraser Hutton, a spokesman for *Symantec UK*, has firmly denied the latest round of scuttlebutt, stating that all extant platforms of *CPAV* would, for the foreseeable future, continue to be maintained and supported. He did confirm, however, that the new *Symantec* anti-virus products for *Windows NT* and for *Windows 95* would go under the name of *Norton Anti-Virus*, although they would incorporate some features currently specific to *Central Point Anti-Virus*.

'Our corporate decision has been to continue to maintain and support *Central Point Anti-Virus*,' said Hutton. 'The product is very popular in the market-place, and has strong customer support. There are absolutely no plans to discontinue its production.' ∎

ESaSS and Reflex Announce Alliance

Following the May agreement between *Norman Data Defense Systems* and the Dutch anti-virus software developer *ESaSS BV* (producers of the *ThunderBYTE!* anti-virus utilities), a further collaboration has been announced between the UK company *Reflex Magnetics* (producers of *disknet*, the security package) and *ESaSS*.

With immediate effect, the two companies will integrate their development teams and pool their technology to build their next generation of anti-virus and security products. Each company, through the agreement, gains the right to market the new products throughout the world, with the exception of 'home territory'.

In a press release, John Buckle, Managing Director of *Reflex*, said: 'By combining the technologies of the two companies, we are set to take the market by storm ... Through tighter integration of our joint technology, *ESaSS* and *Reflex* are set to become the definitive providers of PC security solutions.'

Dick Gehéniau, vice-president of *ESaSS BV*, commented: 'This strategic alliance will translate our technological excellence into increased market share. This closer working relationship is just the beginning. Expect great things.'

Further information on this alliance is available from *ESaSS BV* (Dick Gehéniau) on Tel +31 889 422282, or from *Reflex Magnetics* (Rae Sutton) on Tel +44 171 372 6666 ∎

Virus Prevalence Table - July 1995		
Virus	Incidents	(%) Reports
Form	28	18.9%
Parity Boot	23	15.5%
NYB	13	8.8%
AntiEXE	10	6.8%
Sampo	7	4.7%
JackRipper	7	4.7%
Monkey.B	6	4.1%
AntiCMOS	5	3.4%
One_Half	5	3.4%
Stoned.Angelina	5	3.4%
Junkie	4	2.7%
Viresc	4	2.7%
Leandro	3	2.0%
Bupt	2	1.4%
Stoned.Manitoba	2	1.4%
Stoned.Standard	2	1.4%
* Other	22	14.9%
Total	148	100%

* The Prevalence Table includes one report of each of the following viruses: Amse, Boot.437, She_Has, Cascade-1701, ExeBug.A, Flip, Jerusalem, Jimi, Joshi, K-Hate, LZR, Monkey.A, Natas, Nolnt, Rex, Stoned.Dinamo, Tequila, Tremor, Trojector, Vacsina, V-Sign, and YMP.

VB '95: Boston on the Horizon

From 20-22 September 1995, the *Fifth Annual Virus Bulletin Conference* will be held at the *Park Plaza Hotel* in Boston, Massachusetts. This will be the first time this highly successful gathering has been held in the United States.

The conference key-note speaker is the highly-acclaimed virus researcher, Dr Harold Highland. Many experts will address a wide range of issues, including the susceptibility of *NetWare*, *Windows NT*, *Windows 95* and *Unix* to virus infection, viruses on the Internet and in a corporate environment, and heuristics.

The two-and-a-half day conference will consist of three streams graded according to technical content, and will also feature an exhibition by security soft- and hardware vendors. The partners' programme will feature a tour of the city, and visits to local sites of historical significance.

The fee for the event is £595 (US$895), and *VB* subscribers qualify for a £50 discount. Information is available from the conference manager, Petra Duffield, on:
Tel +44 1235 555139, fax +44 1235 531889 ∎

IBM PC VIRUSES (UPDATE)

The following is a list of updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 21 August 1995. Each entry consists of the virus name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or a dedicated scanner which contains a user-updatable pattern library.

Type Codes

C	Infects COM files	M	Infects Master Boot Sector (Track 0, Head 0, Sector 1)
D	Infects DOS Boot Sector (logical sector 0 on disk)	N	Not memory-resident
E	Infects EXE files	P	Companion virus
L	Link virus	R	Memory-resident after infection

Amazon Queen.468

CER: An appending, 468-byte virus which installs itself in the Interrupt Vector Table. It contains the plain-text messages: 'Amazon Queen...v1.0', 'WHY?' and 'LoRD Zer0'.

```
Amazon Queen.468    E800 005D 81ED 0300 0E1F 06B4 ACCD 213C 3075 0B2E 3B9E D001
```

Amazon Queen.479

CER: An appending, 479-byte variant with the text: 'Amazon Queen...v1.1', 'WHY?' and 'LoRD Zer0'. The first message may be displayed if an infected program is executed and the virus is active in memory.

```
Amazon Queen.479    0E1F E800 005D 81ED 0500 06B4 ACCD 213C 3075 132E 389E DB01
```

Amazon Queen.500

CER: An appending, 500-byte variant with the text: 'Amazon Queen...v2.0', 'WHY?' and 'LoRD Zer0'. The first message may be displayed if an infected program is executed and the virus is active in memory.

```
Amazon Queen.500    81ED 0500 4444 06FF 86F2 01B4 ACCD 213C 3075 132B 3B9E F001
```

Baba.353

CR: An appending, 353-byte variant, named after its 'Are you there?' call: AX=BABAh; Int 21h returns AX=FACCh. It contains the text '=>COMMAND.COM<='.

```
Baba.353            BF00 0181 C646 01B9 0400 FCF3 A45E B8BA BACD 213D CCFA 7503
```

Blue Nine

CR: An appending 925-byte virus with stealth capabilities, which contains the plain-text message: 'Blue Nine Virus by Conzouler 1994'. Of the two known minor variants, B has 'NOP' instructions in its code.

```
Blue Nine.A         50B4 3089 9A02 CD21 81F9 BC01 7466 3C03 7262 8CC3 4B8E C326
Blue Nine.B         50B4 30B9 9A02 CD21 81F9 BC01 7467 3C03 7263 8CC3 4B8E C326
```

Breeder.4026

PR: An encrypted, 4206-byte companion virus which contains the encrypted text: 'File0000.000 = \RENCODES.BRE'

```
Breeder.4206        8D36 1F01 8BFE 8D16 1F01 6D0E 7D0A 2BCA FCAC D0C8 AAE2 FAE9
```

Diddler.91

CNO: A simple, overwriting, 91-byte virus which infects the first file in the current directory. It contains the text: '*.com Diddler 95 (newbee)'.

```
Diddler.91          0AC0 752D B002 BA9E 00B4 3DCD 2193 B95B 00BA 0001 B440 CD21
```

Diddler.190

CN: A simple, appending, 190-byte direct infector with the text: 'Diddler[Newbie] Evolved *.c?m'.

```
Diddler.190         7242 B43F B903 008D 96BE 01CD 213B 8083 BE01 B974 2F3E 8B86
```

Elaine.1127

CER: An appending, 1127-byte virus which contains the text: 'Elaine 1.0 28 May 1994'. As a payload, the virus hooks Int 13h (functions 03h, 0Bh). When active in memory, it may corrupt data in the write buffer (random changes to the first byte in the buffer).

```
Elaine.1127         B813 35CD 2189 9C1B 008C 841D 00B8 FE4B CD21 3D11 1174 4DB8
```

Fistik

CER: An appending, 1280-byte (CDM files) or 1536-byte (EXE files) virus containing the plain-text message 'Dnyalar Tatl', displayed when the virus is active in memory and has infected five files.

```
Fistik              CF3D 004B 7405 2BFF 2E32 012E 803E 3101 0572 03E9 0C02 2E8C
```

Forget.1203

CER: An appending, 1203-byte virus which marks all infected files by putting the byte CCh at the end of programs. In January 1995 it displays the (normally encrypted) message: 'Forget it, I'm lazy today!'.

```
Forget.1203         FCF3 A45E 1F06 B64D 0050 CBB8 43FD BB12 00CD 213D 1256 741A
```

Human Greed.666

ENO: An encrypted, overwriting, 666-byte virus which infects files on drive C. The long message included in the virus body begins: 'That is not dead...' and ends: '...*** HUMAN GREED *** The answer of all evil on earth! Do You Belive? Farwell!'.

```
Human Greed.666     BB2F 018B 1616 01B9 3301 2E31 1483 C602 E803 00E2 F5C3 C386
```

Istanbul.1349

CER: An appending, 1349-byte virus containing the text: 'Anti-Virus??Written in the city of Istanbul (c) 1993' and 'Installed'.

```
Istanbul.1349       3D24 4675 04B8 3434 CF3D 004B 7402 EB6E 5156 5706 5053 521E
```

John

CN: Appending, 1962-byte, direct, fast infector. It displays at random two screens of information on John Buchanan (better known as Aristotle). Infected files start with the plain-text message: 'Ari is a NARC'.

```
John            818E 8A08 4D5A 7437 81BE 8D08 4172 742F 8802 4233 C933 D2CD
```

Maresme

ER: An appending, 1062-byte, encrypted virus containing the text: 'Virus Maresme Show by XUTE !!!'.

```
Maresme         0003 F388 FE8B 9711 0389 8603 AC32 C22A C2CD 01AA CD1C E2F4
```

Milikk

CR: An appending, 1020-byte virus with stealth capabilities, which corrupts the MBS. The virus remembers how often an infected file was executed and keeps the counter inside the MBS of the first hard disk. After 150 infections, it overwrites the boot procedure with its own code. When the system is next started, the text 'M I L I K K' appears in the centre of the screen. After a keystroke, the operating system is loaded as usual.

```
Milikk          E800 005E 88F4 FF81 EE46 04CD 213D 0B00 7503 F972 180E 1F0E
```

Ohlala.1960

CEN: An encrypted, appending, 1960-byte, direct infector which infects six files at a time (three COM, three EXE). It contains the encrypted text: 'Dhhhh La La! Mommmmy, they are teasing me again Shut up you little sonsuvbitches' and '*.MS *VIR.DAT COMMAND'.

```
Ohlala.1960     BB00 002E 8A04 2E30 8129 002E 8A81 2900 89FE 29C6 434E E2EB
```

OS.840

CR: An appending, 840-byte virus which marks all infected files with the string 'OS' placed at the end of programs. It contains only one ASCII string: 'c:\command.com'.

```
OS.840          80FC FF75 03B4 FECF 3D21 2575 01CF 3D00 4B74 03E9 AA01 5053
```

RiP

CR: An appending, 3214-byte virus with the plain-text messages: '>-[RiP]-<' and 'RADICAL_iNVADiNG_PARASiTE (RiP)-ViRUS, iN 94/95 BY AeMlSc, SAYZ Hi 2 U!'. When active in memory, the virus infects an executed COM file and one file in the current directory.

```
RiP             897F 008E 8000 F3A4 C388 8552 CD2F 3D07 0375 03E9 F900 8F39
```

SillyC.140

CN: A simple, appending, 140-byte, fast direct infector. Unlikely to become common in the wild, since it spreads only under DOS 2.11 and when the Country Specifier is set to 2Eh (Sweden).

```
SillyC.140      81ED 0701 8D86 8C01 8F00 0157 A5A5 B438 CD21 3C2E 7512 841A
```

SillyC.190

CN: A simple, appending, 190-byte virus which infects one file at a time. It contains the string: '*.COM'.

```
SillyC.190      A300 018A 45FC A202 01B4 1A81 C782 0088 D7CD 21B4 4833 C981
```

SillyRC.212

CR: A simple, appending, 212-byte virus which marks all infected files by setting the last byte to 0EAh.

```
SillyRC.212     A5A4 C33D 7742 7501 CF3D 004B 756C 5053 5152 1EB8 823D CD21
```

SillyRC.476

CR: Appending, 476-byte virus, similar to SillyRC.212. It contains the plain-text messages: 'Subconsious virus - Conzouler /IR 1995' and 'Mina tankar r det sista som ni tar...'. It also hooks Int 08h and displays for a moment every seven seconds the text: 'LOVE LDVE LOVE LOVE LOVE LOVE LOVE LOVE'.

```
SillyRC.476     4F56 453D 7742 7501 CF3D 004B 756C 5053 5152 1EB8 823D CD21
```

Sofia.432

CR: An appending, 432-byte virus which installs itself in the Interrupt Vector Table. It contains the plain-text messages: 'This Virus is named afler a very nice, clever and cute girl, Sofia', 'Sweden', and 'LoRD Zer0'. The virus creates one hidden, 7-byte long file called 'SOFIA'.

```
Sofia.432       9C80 FC48 743B 3DBE BE74 1D3D 0378 7512 80FF 1975 0D81 FF4C
```

Sofia.528

CR: An appending, 528-byte variant of the Sofia.432. It resides in the same area, contains the same messages and creates an identical, hidden file. It intercepts two more functions (11h and 12h) of Int 21h.

```
Sofia.528       9C80 FC11 742C 80FC 1274 2780 FC48 7473 3DBE BE74 553D 0378
```

Taurus.562

CR: An appending, 562-byte virus containing the encrypted text: 'Happy New Year !' The message is displayed in January, every day between 2:30pm (14:30) and 3:00pm (15:00). The virus reinfects already-infected programs, files growing by 562 bytes with each new infection.

```
Taurus.562      8821 25BA C900 1E06 1FCD 211F 8F14 033E 8803 4747 3E8B 1B47
```

TeaForTwo

CR: An appending, 1024-byte virus containing the plain-text message 'T42 Tea for two!' at the end of infected programs. It was written as a multi-partite virus infecting DOS boot sectors on floppies and files. The copy investigated contains a minor bug, so the virus hooks Int 13h, overwriting some sectors but making diskettes unbootable. The bug is easy to repair, so we will probably see a fix in the near future.

```
TeaForTwo       88FF 25D1 E040 CD21 B425 D0E4 BBFF FFCD 2181 EB60 0084 25D0
```

VCL.279

CNP: A 279-byte companion virus containing the text: '[VCL_MUT] The Pleasure 2 VirusEver have the pleasure?By eMplrE-X'.

```
VCL.279         8903 0051 E808 0059 E2F9 58B4 4CCD 21BA 2C01 E807 00C3 2A2E
```

VCL.316

CNP: A 316-byte companion virus containing the text: '[VCL_MUT] The Pleasure 6 VirusEver have the pleasure?By eMplrE-X'.

```
VCL.316         8903 0051 E808 0059 E2F9 58B4 4CCD 2155 68EC 83EC 40B4 4732
```

Virogen.1535

CER: Polymorphic, appending, minor 1535-byte variant containing the encrypted text: '(c) 1993 Virogen ASeXual Virus v1.00'. It can be detected in memory with the pattern for variant 1520 (see *VB* July 1995).

INSIGHT

Igor Grebert: Carpe Diem

Igor Grebert belongs to a family whose interest in computers reaches back through two generations. He was born on the French Riviera, and grew up in Paris, though he travelled extensively in Europe and the USA. 'Most of my summers,' he said, 'were spent on the beaches around Cannes; sailing, windsurfing, or fishing for sea urchins.'

Family involvement with computers stretches back to the 1970s: 'My uncle and my father designed their own computer called ALVAN in the early 70s. My uncle, Alain Grebert, headed a team of engineers in Philadelphia: they designed a mini-computer around a new language they had developed. It was the first computer I ever programmed - I was eight.'

This exposure led him to the *TRS80* and the *Apple*: games held no interest for Grebert; he was driven to make machines do what he wanted. Later, Grebert studied at one of France's famous engineering schools, *L'Ecole Centrale de Paris*, where he majored in Bio-technology. His special interest was brain simulation: 'In my opinion, there was something missing in the AI field then, and I wanted to understand better what it was.'

Living in America

Grebert fulfilled his military obligations doing research into pattern recognition through neural networks at *Stanford University* in the US: 'I was working with *Boeing*; playing with ideas on making planes land with an improved version of automatic pilots using neural network techniques.'

A few years prior to this, he had met John McAfee, who was at the time working on a PC voice recognition board - Grebert was handling the application programming of the boards in France. This led eventually to a job offer, addressing user interface issues on the *McAfee* anti-virus product.

'That was fun,' reminisced Grebert, 'but after a few weeks there, he challenged me with the Number_of_the_Beast virus, asking me to write a remover for it. That was the beginning of my involvement with PC viruses.'

Then came 512: 'We call it the Stealth,' he said. 'It's kind of interesting to play with a stealth virus at first - I was pretty foolish that time; I was standing there and telling him, "No, John, it doesn't infect, there is nothing, look at it!". That experience made me learn pretty quickly, and I've been learning constantly ever since.'

He still remembers his first encounter with a customer virus problem, a Jerusalem variant which played *Frère Jacques*: 'It triggered a reaction; it was a challenge. 512 was programming; stuff I played with - suddenly, it was affecting customers, people, companies. It was only then I understood that what we were doing was helping - I mean, that company had nothing to do with viruses; it damaged all their backups; made them lose time. They didn't deserve all that.'

The World of Viruses

Grebert has not seen anything really new for over a year now: 'Every new virus we see today belongs to a category which already exists,' he explained. 'This is a contrast to previous years, which makes me think that virus authors are running out of ideas. I believe there will be little change for the next year or so. Then, probably, we will see a few new techniques, but I do not foresee anything radically different.'

Grebert believes that no single anti-virus technique is sufficient to ensure a virus-free environment. Heuristics alone, he believes, will not allow for detection of existing viruses: 'This is why we offer multiple products, and use multiple technologies in our scanners. I believe that we have already integrated the best part of heuristics in our tools and in our scanner, and are now fine-tuning them constantly.'

Heuristics, in his view, have merit, but one must be cautious as to how they are implemented - the inherent risk is false alarm. The future, he feels, is in the harmonious integration of techniques which allow reliable and generic detection of viruses. He sees the best answer to polymorphic viruses as improving virus-specific detection to enable their detection and identification: 'There are simple ways,' he stated, 'to handle these, which are time-effective, and reliable.'

Ethically Speaking

Grebert has definite opinions on virus-writing: 'There is a dilemma between preserving the right of expression and protection against crimes,' he said. 'One should be allowed to play with such ideas as self-replicating code, as long as the environment is strictly controlled, but no-one should be able to force me to run a program I do not want to run on my own machines. Between the two is a fine line which the legal system has yet to define satisfactorily.'

The very thought of virus-writing is alien to Grebert - his only contact with virus authors is through their creations. He has never created a self-replicating program, feeling his time is better spent doing other things: 'The idea of adding the ability to spread has never struck me as interesting,' he said. 'If I have a message, I can use other means to convey it.'

He professes himself disgusted by the amount of time, money, and effort the world has lost over viruses, and does his utmost to counter this, anticipating what the next threat might be, and preparing programs to handle them as soon as possible. 'To do this I do not need to write any such code,' he explained. 'I simply explore the OS internals.'

Igor Grebert is a rarity for a virus researcher; having just as many interests outside work as in!

Professional Growth

Since 1989, Igor Grebert has worked at *McAfee Associates*, an organisation which has recently acquired many smaller companies. Grebert is quick to stress that acquisition played a much smaller role in the deals than development: '*McAfee* is growing out of the anti-virus business towards network management,' he explained. 'Most of our installed base was in companies with networks; people trying to implement anti-virus policies had other problems to address - software distribution, application metering, remote desktop control.

'There are many anti-virus companies around,' he continued. 'It is no longer easy to start a company with no international presence, but new developers can still prove themselves. They have to do this in concert with existing companies, though, as the industry has grown so much. Writing an engine is still fairly easy, and ideas can easily be implemented and tested, but the package is more than the engine.

'You have to support multiple platforms, build interfaces, think network, and client/server. The same thing applies to people who want to write a new OS... What was possible ten years ago is not today - but new opportunities are available today that did not exist then.'

Always, at the core of Grebert's work, are viruses: 'I wanted to work on detection of the "weird" viruses, and... I've always been obsessed with the idea of finding something that would allow me not to work any more. If you're a good programmer, you don't want to waste time, to do things two or three times. One thing you try to do is to automate as much as you can, and to make your scanners as good as possible, so you just push a button to detect the latest virus.

'The technology we had did not allow us to do that - we all have to change some time. What keeps me going at *McAfee* is the opportunity to change technology, and to redesign the scanner from the ground up. As John worked on making the company grow, he allowed me to take on technical leadership; managing the anti-virus researchers and programmers.'

In the Office

Grebert is currently Manager of Research and Development at *McAfee*: 'One of many!' he laughed. 'The anti-virus stuff is what I've been focusing on, but we have network management, we have utilities for *Windows*, we have a replacement for the shell program, and so on...'

Grebert's brief is to find better ways to handle viruses, or to automate the way in which they are processed: 'We retired the older version of our product, and are moving towards a new, more compatible version that goes across platforms, that requires less work from the programmers,' he explained. 'We don't have to rewrite the *Windows* or the *OS/2* parts - it's all integrated, and makes for a very easy-to-use development platform. That was the challenge for our team.'

There are still challenges, however - integrating his knowledge of viruses to a point where the process of detection and removal is almost automatic: 'It's what we have to do! The scanner is the ultimate holder of the technology you've put together. We want the amount of work that has to happen to look at an ordinary virus to be no more than about an hour.

'This is inside a development scheme: you receive the file, someone looks at it, another answers the customer: there's a whole process. The amount of work (granted the virus infects nicely) is a few hours, including removal. When it starts to use techniques which are a little hairier, you need a little more time - but I believe this too can be automated.'

Inside Outside

Though Grebert admits that he was once a 'pizza-and-coke' programmer who routinely worked 80 hours a week, he does now take time out: 'I enjoy going away. I've just come back from Lake Tahoe - it's only a few hours from the Bay, so it's somewhere to go for the weekend. When I travel on business, I often end up spending the weekend in various cities. I like to windsurf - there are places here where I can do that.'

There are still times when he has to work 'from sun-up to sun-down', but Grebert insists that this is not a healthy approach in the long term: 'You cannot do this for four or five years running and still keep your peace of mind.'

Of course, as a Frenchman, one of Grebert's great pleasures in life is food, from sushi to hamburgers ('But you cannot eat hamburgers every day!' he insisted). He enjoys cooking for himself and his friends, and going out to good restaurants: 'There *are* good restaurants here,' he avowed. 'You just have to find them, and be ready to pay the money.'

He does miss France, however; the good food and the cheese (this latter he finds difficult to obtain in the USA) - one day, he says, he will return, but not before his work at *McAfee* is finished. In the meantime, between skiing at Lake Tahoe, and having a house which, in his words, often resembles an international hotel with friends from Australia, Japan, and Europe always around, Igor Grebert remains a man who seizes every day.

VIRUS ANALYSIS 1

What a (Winword.)Concept

Sarah Gordon
Command Software Systems Inc

Winword.Concept is a remarkably friendly virus, which happily infects across platforms. Yes, that's right, *Macintosh*, *MS-DOS*, *Windows NT* - if it runs *MS Word*, it can be infected. Thus, people using mail interfaces which make use of the *Word* application can get a virus by reading electronic mail. The statement 'You cannot get a virus by reading your mail' is no longer true. You can.

Perhaps calling the techniques used by this virus a 'new concept' is not totally accurate. We knew this type of vulnerability in a macro language would be exploited sooner or later. Perhaps we can consider ourselves fortunate that the virus has no destructive payload: its only obvious problem is an inability, in some cases, to save work - it could be worse.

Apparently non-malicious in intent, Winword.Concept nevertheless introduces us to a new threat. In the past, we have seen fast infectors, polymorphics, stealth. This virus merely uses incredibly simple techniques to replicate and hide from the user, once a file is infected.

The appearance of this virus presents anti-virus product developers with a challenge in implementing detection, as, rather than spreading by infecting more traditional types of 'executable' code, it adds itself as a small macro to *Word* templates. This allows the virus to infect and spread utilising files with any extension; as long as they are in *Word* format.

An Operating System by Any Other Name

As applications become increasingly complicated, they have begun to resemble mini-operating systems, supporting their own little file system and command set. *MS Word* has its own programming language, *WordBasic*, which, as the name implies, is reminiscent of 'real' BASIC. Although programming with *WordBasic* is not described in the *Word* manual, further information can be obtained by using the on-line help facilities, or by ordering the *MS Word Developer's Kit*.

Thus, every document has the potential to carry code which represents 'executable' instructions in the *Word* environment. However, this still doesn't explain how these instructions come to be run. After all, even if a document contains a set of macros, they have to be explicitly run, right?

Wrong.

AutoOpen = AutoInfect

In its default configuration, whenever *Word* opens a document, it searches for the presence of a macro named AutoOpen and executes its contents. This is carried out without asking or alerting the user, and so is usually a completely transparent process. The user is aware only that he has successfully opened another document; another triumph of the computer age!

In general, the AutoOpen macro will set up the working environment required by the document or the user. However, *Word* has no concept of privilege and allows the macro to make permanent changes to the way it functions. This is a powerful and useful feature, and one which is open to a great deal of misuse.

In the case of Winword.Concept, the AutoOpen macro first checks to see if the virus is already active on this computer, by searching the environment for the presence of a macro named 'PayLoad'. If this is present, execution aborts.

A second check is made for the presence of a macro named 'FileSaveAs'; if found, the virus sets an internal flag, and again aborts infection. The internal flag used by the virus to signify this is called 'TooMuchTrouble', possibly indicating that if the user already has a macro named 'FileSaveAs', it is simply too much trouble to continue and infect the system.

If these tests are passed, the virus adds four new macros to the user's 'global document template'. This is stored in a file named NORMAL.DOT, and is a general purpose template for any document.

To quote from the *Word* manual: 'Unless you select another template when you create a new document, *Word* will base the document on the Normal template.' The four new macros are AAAZAO, AAAZFS, PayLoad and FileSaveAs (the contents of the FileSaveAs macro are simply copied from the virus' macro AAAZFS).

The virus displays a dialog box upon infection, containing what appears to be an infection counter, but which displays the number '1' no matter how many infections you generate. On examination of the macro code, it is observed that this is due to sloppy programming on the virus author's part.

Once this message box is clicked on, the virus is resident, and execution of its 'bootstrap' macro finishes. Once resident, the virus code is activated whenever the user attempts to save a file using 'File/Save As', as this function has been 'enhanced' by the addition of a FileSaveAs macro. Whenever the user selects this option, the virus creates an AutoOpen macro in the new document, and copies the contents of the macro AAAZAO into it. The macros AAAZFS, AAAZAO and PayLoad are also created and copied into the new document.

Thus, the virus code is added to all those documents which are stored using File/Save As, and it is ready and waiting to spread when that document is sent to another unsuspecting

user. There are two things worth noting: the macro called 'PayLoad' is never executed, and it contains only the following text:

```
Sub MAIN
    REM That's enough to prove my point
End Sub
```

The name of this macro is not an empty threat: examination of the virus code and the *WordBasic* language shows that it would require a trivial alteration to make the PayLoad macro active and to give it a wide variety of different functions.

Detection and Removal

Checking whether a copy of *Word* already contains the virus is trivial. Start the program, and select the Macro option under the Tools menu, choosing Macros Available in 'All Active Templates' option.

This displays a list of macros currently installed on the computer; if AAAZAO, AAAZFS, FileSaveAs, and PayLoad are present, the machine is infected. Highlight each of the virus' macros in turn and select the Delete option. This removes the virus, but does not solve the problem of the infected files on the system.

There are other ways to detect this virus in files. One is to add user-defined virus strings to anti-virus programs which have this feature. The user can add '3A 41 41 41 5A 41 4F' and/or '3A 41 41 41 5A 46 53', scanning all files. These scan strings are the hex representation of the ASCII strings ':AAAZAO' and ':AAAZFS', and will be found in any document containing that text.

Since .DOT and .DOC files are not typically scanned, it is important to remember to add them to the list of file types to be scanned. If you suspect you have this virus, you may want to scan *all* files, as your users may have changed the filename extensions after saving the files.

Alternatively, you can search every document on your system for the strings (and the rest of the virus) using a disk editor. This could prove a lengthy process and is not recommended.

If you find these strings in a *Word* document, further checks must be made. Unfortunately, these are difficult, as the virus is composed entirely of plain text, making it difficult for someone without knowledge of *Word* to decide whether even a *Word* document which contains these text strings is the virus itself, or a message warning of the virus' presence.

One definitive way to determine whether the document is infected is to open it using *Word*, though this is counterproductive. My suggestion is that if you find the macros listed above active within *Word*, call your anti-virus software vendor, who should be able to talk you through a fix.

You can restore infected documents to their pre-infected state manually. To do this, with your infected document loaded, do the following:

- use Edit/Select All to mark the whole document; then Edit/Copy to copy the document to the clipboard
- create a new, untitled document using File/New
- using Edit/Paste, place the contents of the clipboard into the new document
- close the original document using File/Close
- if you are certain that the new document is identical to the old, except for the missing virus macros, use File/Save (*not* File/Save As) to save the new document over the old
- if you are not certain the new document is identical to the old, use File/Save to save the new document with a new name, keeping the infected document isolated in a safe place until you are sure you no longer need it

Manual removal of the virus via other methods is best performed by someone experienced in *Word* document structure.

Automated detection and removal of the virus is offered by several vendors, including *Command Software Systems*; its fix, Wvfix.zip is available free of charge from the *Command/F-Prot* library section of the *NCSA* Anti-Virus vendor forum on *CompuServe,* or via anonymous FTP from ftp.commandcom.com (questions/comments may be mailed to winword@commandcom.com, and will probably end up in my mailbox).

The Problem; the Solution

The techniques used by this virus are so simple that any idiot could use them to construct similar viruses. If history is an indicator, we can expect to see more of this type of virus.

While a short-term fix is available, the ease of creation and modification means that we must find a long-term solution to this general threat. As far as I can see, the most likely way will be to alert the user to any changes made to his global settings. While this will not prevent such a virus from spreading, it will provide users with some warning before their application is reconfigured.

Security is no longer the realm of the OS developer; application programmers should keep a careful eye on the possible misuse of the extra functionality they are providing.

Winword.Concept

Aliases:	Word prank macro.
Infection:	*MS Word* documents.
Self-recognition in *MS Word* documents:	
	Searches for a macro named 'PayLoad'.
Trigger:	None.
Removal:	See text.

VIRUS ANALYSIS 2

Byway: The Return of Dir_II

Dmitry O Gryaznov
S&S International plc

Those people who have been interested in computer viruses since the early 1990s may remember the 'pancomputeria' caused by the Dir_II virus in the autumn of 1991 - this was a virus which swept around the world like wildfire.

History of the Technique

An infection technique which was completely new at that time was introduced with the advent of the Dir_II virus, and made it the fastest infector ever. In fact, Dir_II brought with it a completely new category of computer viruses: file system infectors.

The virus installs itself as the main DOS disk driver, and intercepts all disk accesses to floppy or hard disks. Then, on any disk access, Dir_II scans the data being read or written for possible disk directories.

If the data reveals a directory, the virus modifies all directory entries referring to executable (COM/EXE) files to point to one and the same cluster chain where the virus has stored its body. The original start cluster number of an infected file is stored, encrypted, in the unused parts of the DOS directory entry.

When the virus is memory resident, everything appears normal, since the virus intercepts any directory accesses, modifying the images of directory entries in memory to their condition before infection.

When there is no virus in memory, however, DOS 'sees' the actual state of directory entries as they are stored on the disk. In this case, since all the executable files are cross-linked to the same cluster, running any executable file results in the virus being loaded to memory and executed.

Strictly speaking, Dir_II does not infect files - the file data, as well as its cluster chain, remains unchanged. The virus 'infects' directory entries instead, cross-linking them to the single cluster chain containing the virus body. So, if you boot a computer from a clean DOS diskette and run CHKDSK on an infected disk, CHKDSK will report dozens of files cross-linked to the same cluster, as well as dozens of lost cluster chains.

With the virus in memory, however, everything looks fine. Since Dir_II intercepts disk accesses at a DOS driver level, presenting itself as the main DOS built-in disk driver, just about any disk access will enable the virus to replicate. Simply typing DIR is sufficient to enable the virus to infect all the executables in the directory from which you requested a listing.

If you accidentally type WIM instead of WIN, DOS will look for an executable file named WIM.COM (or WIM.EXE, or WIM.BAT) not only in the current working directory, but in all the directories listed in the PATH environment variable as well. The result is that all the executable files in each of these directories will be infected by the virus.

This infection technique enabled Dir_II to propagate with unparalleled speed. First released in Bulgaria, it took Dir_II only several weeks to become the most widespread virus in the world in the autumn and winter of 1991.

Fortunately, it did not last long. Dir_II is now believed to have been extinct in the wild for some time, mainly because it appeared to be incompatible with DOS versions 5.0 and above. The memories of this virus survived, making Dir_II a sort of anti-virus 'scary legend'. Yet recently we have faced a 'reincarnation' of Dir_II, in the form of a virus called Byway or TheHnd.

"unlike Dir_II, however, Byway operates pretty well even under the latest versions of DOS"

Dir_II Reincarnate

Byway uses the same extremely fast and effective infection technique which was introduced in Dir_II. Unlike Dir_II however, Byway operates pretty well even under the latest versions of DOS, a fact which might well make it the Dir_II nightmare of today.

To make things even worse, Byway is a polymorphic virus, changing its appearance from one infected disk to another. Its code is written in an extremely obfuscatory manner, with many self-modifying instructions and unusual addressing modes. All this helps make its disassembly and analysis anything but a piece of cake.

Stealth Capabilities - Not Quite There

Still, there is a flaw in this otherwise next-to-perfect virus: its stealth capabilities. To protect the cluster chain where the virus body is kept, Byway creates a 2048-byte-long file called CHKLISTx.MSx in the root directory of an infected disk. The character 'x' in the file name represents the non-printable ASCII code 255 (0FFh), which is displayed onscreen as a space.

The file has System, Hidden and ReadOnly attributes set, so it cannot be viewed by a simple DIR command. You can, however, use the DIR command '/ASH' to see the file. The

switch '/A' forces DIR to show files with particular attribute bits set; the switch '/SH' specifies System and Hidden respectively. So, if you see a file called 'CHKLISTx.MSx' with these attributes, your computer is likely to be infected with Byway!

Text Strings and Trigger

In other ways, the virus is functionally very similar to Dir_II, although, judging by its disassembly, it was an independent 'project'.

The text strings: '<by:Wai-Chan,Aug94,UCV>' and 'The-HndV' are found inside the encrypted virus body. The former, slightly altered, gives the virus its name of Byway, though variations on the first (TheHnd) are also used.

Starting in 1996, providing the day of the month is equal to the doubled month number plus two (i.e. 4 January, 6 February, ..., 26 December), the virus may trigger while infecting a computer.

When triggered, Byway displays a scrolling text phrase, 'TRABAJEMOS TODOS POR VENEZUELA!!!', accompanied by a tune which might well be Venezuela's national anthem. The phrase itself is Spanish for 'Let us all work for Venezuela!!!' or something close to it - I do not speak Spanish myself, alas.

We at *S&S International* are currently receiving an increasing number of technical support calls regarding Byway. Unfortunately, they prove the prediction that the virus is quickly becoming very widespread - exactly like its forerunner, Dir_II.

Detection and Repair

Fortunately, several anti-virus products are already capable of detecting this virus. As for repair, the method used to remove Dir_II also works well with Byway. This is, basically: 'Let the virus disinfect itself', a strategy which works not only for file system infectors, but for full-stealth viruses as well.

The removal method is based on the fact that a stealth virus effectively 'removes' itself from a file being read. The word 'removes' is in quotes because a virus does not necessarily remove itself physically from the file, but rather returns the image of the file in memory to the condition in which it was before infection.

So, if an infected file is copied to a place which a stealth virus cannot infect while the virus is active in memory, the copy will be virus-free. In the case of both Dir_II and Byway, it is enough to PKZIP (or ARJ, LHA, etc) all the files on an infected disk while the virus is active in memory, then boot from a clean system diskette, reformat the disk, and restore the files from the archive. Due to Byway's stealth technology, file copies which are placed within the archive will be disinfected.

Also, since the virus infects at the DOS driver level, it is not able to infect any files on a *Novell* (or, for that matter, any other) network file server. So, it is possible simply to copy all the files from an infected workstation (whilst having the virus active in memory, mind you!) to a server, reboot the workstation from a clean DOS floppy disk, reformat the local hard disk, then restore all the files from the server to the workstation.

The third possibility would be to back up the contents of an infected disk to a tape on a 'dirty' machine and to restore them to the reformatted disk in a virus-free environment.

There are at present two slightly different variants of Byway known. They contain somewhat different encrypted text messages, but are functionally virtually identical. Therefore, both detection and disinfection methods described above will work for either of the two variants.

Byway

Aliases: DirII.TheHnd, DIR2.BYWAY, DIR.TheHnd.

Type: Polymorphic, memory-resident, encrypted file infector with stealth capabilities.

Infection: All executable files.

Recognition:

The DOS command 'DIR /ASH' shows a 2048-byte-long file called CHKLISTx.MSx, with System, ReadOnly, and Hidden attributes set, in the root directory of the infected disk.

Self-recognition in Files:

Compares the starting cluster number to that of the virus.

Hex Pattern in Files:

```
8BF0 * 8BFE * FD * 4974 *
AD * 35 * AB * EB
```

(within 28h bytes of beginning of file)

Hex Pattern in Memory:

```
501E 5657 B0F0 BE58 040E 1FFC
06C4 7C1B A4A5 A58B 7C1B A407
```

Intercepts: No interrupts intercepted.

Trigger: Running text message displayed: 'TRABAJEMOS TODOS POR VENEZUELA!!!', accompanied by tune.

Removal: PKZIP (or similar) all files on infected hard disk, boot from clean system floppy, restore hard disk, and restore files from archive.

VIRUS ANALYSIS 3

Rainbow: To Envy or to Hate

Jakub Kaminski

Only a small number of the thousands of viruses written merit analysis. Most researchers do not have the time to go through even those which are 'worth' examining closely. Often, when a virus is detected and cleaned, it is shifted to the 'to-do-in-near-undefined-future' pile. Those which do encourage closer examination are likely to be new, unknown specimens spreading quickly in the real world.

Not long ago, I was asked to check a PC which could no longer run *Windows*, and had problems booting from a floppy. I expected to find corrupted files or sectors, along with disabled boot from floppy, or perhaps something 'Monkey-like' fiddling with the partition table data.

My investigations revealed a 2351-byte, multi-partite virus spreading through partitions and directories, residing in the boot sector and many executable files. Its most interesting characteristics are its stealth techniques, and the method by which it disables clean boot from system floppy without altering the contents of the CMOS. An attempt to start from a system diskette results in a system hang before a command prompt appears - neither drive C nor drive A is accessible.

Infection Symptoms

This virus, Rainbow, infects the MBS of hard disks, DOS boot sector of floppies, COM files, and files with EXE-type structure (EXE, DRV, 386, XTP). It is unencrypted, and named afler a plain-text message inside its body: 'roy g biv' (an acronym of the colours of the rainbow).

The virus attaches itself to the end of programs. All infected programs have their time stamp modified; the field containing the number of seconds divided by two is set to 31. On infecting a DOS boot sector, Rainbow changes only 25 bytes at offset 3Eh, adding a jump instruction at the sector beginning. The copy of the original boot sector is kept in the diskette's last sector, and the remainder of the virus code written in the preceding five sectors.

When the MBS is infected, only its initial 25 bytes are changed by the virus. The rest of the virus body is written into five sectors on track 0 (cylinder 0, head 0), starting from sector 2. Rainbow does not keep a complete copy of the MBS: the 25 bytes it replaces are stored in sector 6, offset 142h. It also modifies the MBS in a way which could be described as self-protection or as the payload itself.

The information on the active partition (16 bytes) is copied to sector 6, offset 132h, and the contents of the original Partition Table replaced by this Hex byte sequence:

```
0000 0100 0500 B80B 0100 0000 BC01 0000
```

This is interpreted by the operating system as a non-active, extended DOS partition, starting from head 0, cylinder 0, sector 1; ending on head 0, cylinder 523, sector 56; beginning one sector from the start of the disk, and containing 444 sectors in total. The most important characteristic is that this partition entry points not to another partition but to the MBS itself (head 0, cylinder 0, sector 1). Such a case is often referred to as 'the recursive partition' and can be a big headache to someone using the latest versions of *MS-DOS*.

For users of v5 or v6.x of *MS-DOS*, access to the system containing the recursive partition is no longer possible. Starting from a hard disk or a diskette will put the system in an endless loop in the middle of the boot sequence (the OS loader traces through the extended partition chains and locks itself up, investigating the same sector again and again).

Rainbow incorporates a significant number of system control and stealth procedures. When active in memory, it hooks interrupts 01h (anti-debugging), 12h (hiding 'missing' memory), 13h ('Are you there?' call, stealth/infection of boot sectors), 21h (14 functions used for stealth/infection of files), 24h (stealth), and 2Fh (stealth).

Execution of Infected Files

When an infected file is executed, the virus checks to see if the system is infected, and whether the virus is active in memory. This is done by issuing an 'Are you there?' call (Int 13h, AX=1BADh). The value DEEDh returned in the register AXh means the virus is in control [*'One bad deed', geddit? Ed.*], in which case the original program is restored in memory and its execution follows in the usual way.

If the system is clean, the virus installs itself in memory. It takes the 3K required from the current block of memory (as long as it is the last one in the memory block chain), usually placing its code 3K below the current top of memory. Since the virus relies on the data in the current PSP, it will install itself above the 640K limit if an infected file is loaded high.

Next, the virus hooks Int 01h, and tries to install its own Int 21h handler. Rainbow changes not the Interrupt Vector Table, but the current Int 21h service routine. Installation takes place only if the current Int 21h procedure begins:

```
CMP AH,??
JNBE ??
```

The virus replaces these instructions with a FAR JUMP to its own code, saving the original pointers in the virus code. Then, it hooks Int 2F and installs its Int 13h ('Are you there?' call, response only) handler.

Now, the virus infects the MBS of the first physical hard disk. The Int 13h service routine is modified to include full stealth procedures. Int 12h is then intercepted and a new

procedure installed which hides the 'missing' memory occupied by the virus. Finally, the infected file is restored in memory, and control is passed to the original program.

Booting from an Infected Disk

When the code in the infected boot sector is executed, the virus locates the top of memory, decreases it by 3K, and copies all of its code into the area allocated.

Now, Rainbow installs its Int 13h handler (with all infection and stealth features). This also includes the code to install its Int 21h handler after the rest of the operating system is loaded. The virus relies on checking the address of the Int 24h service routine. If its segment is smaller than 1000h, the virus assumes that DOS is already loaded.

In Memory

When an infected file is executed, Rainbow installs itself in memory, intercepting all subsequent interrupts. Unlike most multi-partite viruses, it does not have to be loaded from an infected boot sector to gain full functionality. Rainbow can spread and infect files and floppy boot sectors even on workstations with no hard disk.

The virus infects diskettes on Read or Write access. When active in memory, it returns the clean, original sector at each attempt to read the DOS boot sector. Files are infected on Execution (Int 21h, function 4Bh), or when opened.

COM-type files are infected only if they are less than 63057 bytes and their extension is COM or com. EXE-type files are infected when file length is as specified in the EXE header. Rainbow's stealth procedures include hiding the length of infected files and the virus signature in the file time stamp.

As self-recognition in files is based on the time stamp, attempts to execute a clean file with a time set to 62 seconds often results in a system crash: the stealth procedure tries to disinfect a clean file, but corrupts it instead. It is the only serious bug (minor, in comparison to the poor coding in the vast majority of viruses) which I found in its code.

Booting Clean

The safe removal of any virus from an infected system is always based on a clean boot from a system diskette, something which, in this case, is not always easy. Those still using *MS-DOS* v4 or lower can use the usual system floppies, but those who upgraded to v5 or higher may find themselves in trouble if Rainbow infects their machines.

To gain access to an infected/corrupted MBS, eradicating the recursive partition problem, either boot from an older version of DOS, or boot from an infected disk, then disable the virus in memory or avoid its stealth routines.

If the former is chosen, a system floppy which has an older version of DOS is required - but how many laptop users have a bootable, 3.5-inch DOS 4 diskette? Diagnostic diskettes which boot to their own operating systems can also help in gaining access to a disk which has a recursive partition problem.

The latter solution requires an anti-virus product which can detect and disable viruses in memory, or can work properly when viruses are active in the system. In the case of the Rainbow virus, this does not appear to be a simple task.

Conclusion

One of the plain-text messages inside the virus body is: '*4U2NV*', which can be read as: 'For you to envy'. Some virus writers may certainly envy the author of Rainbow his ideas and skills, but if this virus becomes common in the wild, the majority of the PC community will only hate him.

Rainbow

Aliases: None.

Type: Multi-partite, stealth, COM/EXE/MBS/DBS infector.

Self-recognition:

MBS: word 83A5 Hex at offset 15h.
DBS: word 83A5 Hex at offset 53h.
Files: seconds field in time stamp = 62.

Hex Pattern in MBS:

```
BB00 7C8E D38B E38E C3B8 0502
B902 00BA 8000 CD13 9AA5 8300
```

Hex Pattern in DBS:

```
BB00 7C8E D38B E38E C3B8 0502
B9?? ??BA 0001 CD13 9AA5 8300
```

Hex Pattern in Files and Memory:

```
E800 005E 83EE 03B8 AD1B CD13
3DED DE75 450E 1F81 C664 0781
```

Intercepts: Int 01h, anti-debugging; Int 12h, hiding missing memory; Int 13h, boot sector infection/stealth; Int 21h (functions 11h, 12h, 3Ch, 3Dh, 3Eh, 3Fh, 40h, 42h, 4Bh, 4Eh, 4Fh, 57h, 5Bh, 6Ch), file infection/stealth; Ints 24h/2Fh, stealth.

Trigger: Recursive partition in infected MBS.

Removal: MBS - boot clean from DOS 4 or lower, replace first 25 bytes with the bytes from sector 6 offset 142h, replace recursive-partition data with 16 bytes from sector 6 offset 132h. Alternatively, boot from infected hard disk and disable virus in memory before repairing MBS. Files - although cleaning infected files is relatively easy, to remove virus safely, repair MBS, boot clean and replace infected files with a clean backup copy.

TUTORIAL

Circular Extended Partitions: Round and Round with DOS

Mike Lambert

On pages 12-13 of this month's *VB* is an analysis of Rainbow, a virus *VB* first mentioned in its July 1995 *IBM PC* Viruses (Update). The entry states: 'A system with an infected MBS cannot be booted from a clean system floppy if the machine is running any DOS version of 5.0 or higher'. When the virus was brought to my attention, I thought of the paper I co-wrote with Charlie Moore, 'Circular Extended Partitions: A DOS Vulnerability' (December 1992).

Recognising the Problem

The symptoms of a circular extended partition can be described as follows: when booted, the operating system load hangs and the hard disk access light stays on steadily. The kernel is hung in a loop, reading the same block (or circular chain of blocks) from the hard disk. The solution is to boot a version of DOS without the bug in its kernel.

In the paper mentioned, I published patches for DOS 3.3-5.x (a single-byte patch for each). *IBM* sent me each version of *PCDOS* and asked me to publish a patch for each. *DRDOS* was too complex to patch, so was omitted. *MS-DOS* patches were included in case they were needed in an emergency.

More information on the circular extended partition problem, and a tutorial on DOS disk structures, is included in the paper mentioned above, 'Circular Extended Partitions: A DOS Vulnerability', by Mike Lambert and Charles Moore.

The Rainbow Virus

Rainbow implements the simplest of circular extended partitions. It replaces the entry describing the bootable DOS partition in the Partition Table with a phoney extended partition which points to the MBS. The virus 'stealths' the MBS reads so that, when the virus is resident, DOS sees the correct DOS partition entry and the OS comes up normally. When the virus is not resident, DOS versions which have the circular extended partition bug will hang when booted.

The circular extended partition in Rainbow does not hang *MS-DOS* v3.3 or v4.01 - these can be used to boot today's systems (Rainbow does not work on older CPUs) in the event that an *MS-DOS* v5 or v6.x system does not boot.

To remove the virus, it is necessary to clean-boot a version of DOS which does not have the bug, then restore the MBS from a backup copy. The system should then be rebooted from the floppy (so that DOS will see the DOS partition), and infected files should be replaced.

Circulating a Fix

While circular extended partitions were a problem for all *Microsoft*, *IBM*, and *DRDOS* versions implementing extended partitions until December 1992 (v3.3-v5), the issue should pose no problem to the latest versions - Charlie Moore and I notified all three operating system developers in September/October 1992.

Our paper identified a coding error which results in the problem (this was confirmed by *IBM*). *IBM* and *DRDOS* were happy to hear about the problem, and promised to correct it in the next version.

Microsoft proved to be difficult to contact and did not return calls, faxes, or a message on the *MS-DOS 6.0* beta test hotline. A subsequent article by another author brought the problem more directly to *Microsoft* technical staff via the Public Relations office.

DOS Version 6.x

Curious to explain the note in July's *VB*, I assembled v6 of *MS-DOS* and *PCDOS* products and did some testing. True to their word, *IBM* had corrected the problem in *PCDOS 6.1* (no problem with *PCDOS 6.3* either). Testing the *Microsoft* version 6 series explained the note.

Microsoft v6.0, v6.2, v6.21, and *v6.22* all still have the same bug in IO.SYS, meaning that *MS-DOS* v3.3 to 6.22 (*PCDOS* v3.3 to 5.02, and *DRDOS* v6.0) will not boot in the presence of a circular extended partition. *IBM* v6.1 and v6.3 do not have the bug. As I have been unable to test with the latest version of *DRDOS*, I do not know if the problem has been corrected as yet.

MS-DOS 6.x Patches

The only responsible thing to do is to publish the patches for the *MS-DOS 6* series in case there should ever be a need to recover an *MS-DOS* system from such a problem. The patch is exactly the same for each version of *MS-DOS 6.x*. Within IO.SYS, the procedure is:

1. Search for bytes 07 72 03 - these are at offset 2918h.
2. Change 03 at offset 291Ah to 06.
3. Write the change back to disk.

I have tested each patch, and all work as intended. The decision to use the patch to bring up a system crippled with circular extended partitions lies with the individual.

If Rainbow ever makes it into the wild, it might be a good idea for *MS-DOS* users to have a disaster recovery floppy without the bug (*IBM v6.1* and *v6.3* do not have it) until *Microsoft* applies fixes to *MS-DOS*.

FEATURE

Computer Viruses: Naming and Classification

David B Hull, PhD
National University, California

The literature of computer viruses is steeped in biological analogy. Even their choice of name, virus, is a direct analogy to biological organisms. The writers of this pernicious code also use this analogy: witness the Dark Avenger's Mutation Engine. Indeed, some parts of the community have gone so far as to suggest the concept of artificial life for these and related creations [*Ludwig, 1993; Stojakovic-Celustka, 1994*].

This paper is an extension of the analogy to the problems of naming and classifying computer viruses. These two issues are problems which are critical to working with living and non-living creations. The need for precise name and classification is rooted in the need to communicate effectively about the item in question. This is true regardless of whether the creation is man-made, such as a Mozart sonata, or natural, such as a lemur.

Naming

Naming involves the development of a set of protocols for creating an acceptable name for any given item in the set under review. The more universally accepted the naming protocol, and the more widely it is used, the more valuable it will become.

Modern zoology has benefited greatly from the adoption of a uniform code: *The International Code of Zoological Nomenclature* [*ICZN, 1964*]. This is a remarkable work, and I recommend it as a model of solutions to issues faced by current virus and anti-virus researchers. It derives from the work of Linneaus, the 'father' of modern biological nomenclature, and in particular is founded on the tenth edition of the *Systema Naturae* published in 1758.

The *ICZN* presents several underlying principals which need to be addressed. First, following Linneaus, it uses a binomial nomenclature; that is, a genus and species name together identify an animal. This can be supplemented as needed with names for Family, Order, etc. However, the Code does not define exactly what a species or genus is.

Second, it establishes a protocol for creating and emending zoological names, which in this case are in Latin or pseudo-Latin and Greek or pseudo-Greek.

Third, it uses the rule of priority (i.e. that the chronologically earliest-recorded name will take precedence) to impose order among conflicting claims about the correct name. The *ICZN* has developed and refined this naming framework. The exact requirements for a valid publication of an *ICZN* name are beyond the scope of this work, but they are certainly worth studying.

Fourth, it ties the name of the species, or genus, to a type specimen. The code does get rather involved here, because this concept is critical to the whole naming process. The important points to note are that the name is tied to a particular specimen, and that this specimen is available to other professionals in the field to examine and compare with other material.

The types must be deposited in a museum or other institution: 'Every institution in which types are deposited should (1) ensure that all are clearly marked so that they will be unmistakably recognized; (2) take all necessary steps for their safe preservation' [*ICZN, 1964*].

Naming protocols are, however, basically independent of a commitment to an underlying organizational structure of the organisms being studied. Indeed, Linneaus had no particular underlying philosophy about the mechanisms and organizational structures underlying what he named [*Hull, 1973*].

Classifying

Classification involves grouping the items in the set under review into categories. In many cases, such as zoology, these categories are nested hierarchically. Classification does involve an underlying philosophy about the mechanisms and organizational structure of the items and groups being classified. This philosophy is also strongly influenced by the purpose for which the classification is to be used.

The division between phenetic, or structural, classification and phylogenetic, or evolutionary, classification has a long and deep history in zoology, for example [*Heywood & Mcneill, 1964*]. The classification of Shakespeare's works for library retrieval as contrasted with literary analysis to determine authorship provides an even starker contrast.

Basically, classification approaches may be divided into three categories: heuristic or morphological groupings aimed at simple assessments of similarity; phylogenetic groupings aimed at tracing evolutionary relationships; and functional classifications grouping by categories of action.

Classifying a killer whale *Orcinus orca*, a gray wolf *Canis lupus*, and a great white shark *Carcharodon carchiaras*, must produce very different groupings with each approach. Gross morphology might group the whale and the shark together as torpedo-shaped sea animals, in contrast to the wolf. Phylogeny clearly would group the whale and the wolf together as mammals against the shark. Functionally, all three are high level carnivores!

Beyond the question of the philosophy underlying the grouping is the issue of actualization. The type of data to be gathered, and the means of developing groupings from the data, is critical to the success and usefulness of the classification schema. Information needed for developing evolutionary relationships is often not available directly and must be inferred from other data. Even the choice of method to create groups can have significant impact on the results; viz the differences in different mathematical clustering techniques on the same similarity matrix data. These choices affect the usefulness of the classification system.

Creating a field guide relies on distinctive features used in the grouping methodology. This holds true whether it is monkeys or missiles being identified.

Current Naming and Classification of PC Viruses

Let us start by examining how viruses are currently named, using the results of the naming committee of the *Computer Anti-virus Research Organizations* [*CARO, 1991*]. *CARO's* classification follows a hierarchical format, based on structural similarity of virus code [from *ftp.informatik.uni-hamburg.de:/pub/virus/texts/tests/vtc/naming.zip - 8/20/94*].

The virus name consists of four parts (to be discussed further in *VB*, October 1995), delimited by periods. The underlying classification scheme is explicitly stated to be based on 'structural similarities of the virus' [*CARO, 1991*].

> *"the transplantation of ... code from one virus to another need not represent an evolutionary relationship"*

An important component of similarity is the use of identical sections of computer code in similar viruses. This can come about either because actual sections of code have been copied from a previous version of a virus, or because similar functionality leads to similar code. The use of structural similarity is not absolutely enforced in the *CARO* scheme.

A second major consideration is the length of active code. 'All short (100 bytes of code or less, messages excluded) overwriting viruses are grouped under a Family_Name, called Trivial. The variants in each family are named by their infective length' [*CARO, 1991*].

Functional criteria (resident versus non-resident) and the type of file infected (COM, EXE, MBS or boot sector) also play a part. In an effort to fit all the viruses in the scheme, classification categories for viruses written in high level languages are also represented by a separate category.

The *CARO* effort is clearly aimed at providing a solid and stable naming system for virus-scanning software. However, the exact methodology used to create *CARO's* classification has never, to my knowledge, been presented publicly.

Naming Issues

A major problem in the current nomenclature of computer viruses revolves around the lack of widely-accepted standards. This leads to many communication problems. Perhaps the most obvious (and also perhaps the most amusing) is *McAfee's* 'Genb' and 'Genp' virus - this is their shorthand notation for a generic boot sector virus and a generic hard disk partition virus.

Virus names should consistently and unequivocally name a specific computer virus. The *CARO* scheme is an excellent discussion piece for developing such nomenclature; however, it should be based on structural similarity only. Other considerations, such as mode of action, or the language used to write the virus, are not central to identification.

Furthermore, as Spafford rightly recognizes, the mode of action and programming language used will mark the structure of the resulting machine code strongly, in any case [*Spafford & Weeber, 1992*].

The second major issue in naming involves specifying exactly what the name represents. Unless the name of a virus is specifically linked to a known piece of code, it is never clear precisely what is being discussed. This leads to the type concept used by the *ICZN*.

In zoology, each scientific name is directly linked to a museum specimen, or other known identification of the organism. This is called the type specimen. It is usually held in a museum, and is specially identified as a type, holotype, lectotype, etc, depending on its exact relationship to the name it represents. These type specimens form the key identifiers for a given name. A group takes its name from that of the type specimen with which it is classified.

The third major issue involves establishment of a valid name. The *ICZN* establishes valid names by 'priority of publication'. In its simplest form, this means that the earliest publication of a valid type description of a previously undescribed organism establishes its name.

This requires demonstration that the new specimen is 'different' from all previously known specimens, creation of a valid name under the *ICZN* rules, designation of the type specimen, and publication in a responsible journal. If, on re-examination, the specimen is found to belong in the same group as an organism with another valid name, the earlier of the two names applies to this group.

Classification Issues

There is also a problem of phenetic (structural) versus phylogenetic (evolutionary) classification. In biological classification, the ultimate goal is to develop an understanding of evolutionary, or phylogenetic, relationships. All classifications still begin with phenetic or structural similarities. These phenetic characteristics are weighted to reflect their relative importance as phylogenetic indicators [*Jardine & Sibson, 1971*].

There is a great deal of confusion in virus classification as to the goal of classification. At one extreme, *CARO* is concentrating on recognition of computer viruses - not an unreasonable approach for an anti-virus organization. In many ways, it parallels the approach used by classical zoological taxonomists, and popular field guides to animals.

At the opposite end of the spectrum are classifications focusing on the evolution of computer virus techniques, and on the individuals writing computer viruses.

Bontchev's discussion of the Bulgarian and Soviet virus 'factories' is a classic in this approach [*Bontchev, 1992*]. Gilad Japhet's anti-virus program *CORAL* appears to be developing towards an evolutionary approach, using techniques which appear to be similar to analytical approaches used in this paper [*Japhet, 1994*].

Computer viruses evolve in complex ways not usually encountered in nature. The transplantation of large segments of computer code from one virus to another need not represent an evolutionary relationship, for example. A newer virus may just represent a debugged or patched earlier version. The virus author may have deliberately incorporated parts of other viruses as a short cut, or because the plagiarized code is useful.

If the virus incorporates code generating 'engines', similar code may appear in viruses with no other similarities. Structural similarities deriving from functional similarities likewise derive from several sources.

There are only certain ways to do certain things with a PC running under DOS, for example. Programmers also, like writers in general, have a particular individual style which leads to coding similarities.

Spafford uses the example of the Internet Worm, where the code used linked lists as the primary data structures. It seems that the first class on data structures and algorithms which Robert T Morris took as an undergraduate used LISP: the lesson stuck all too well [*Spafford & Weeber, 1992*]. This makes using zoological concepts such as 'parallel evolution' particularly tricky in analyzing computer viruses.

A second, tricky problem involves defining the unit of classification. In zoology, the essential unit is the species. A phenetic (structural) definition of a species specifies the smallest statistically coherent unit [*Jardine & Sibson, 1971*]. The phylogenetic (evolutionary) definition of a biological species based on the capability of interbreeding does not appear to have much relevance to computer viruses.

Interestingly, a recent article has presented the idea that the definition of a biological species does not have much application to living viruses, either [*Eigen, 1993*]. This article presents the concept of a 'quasispecies', which Eigen describes as: 'a multitude of distinct but related nucleic acid polymers. Its wild type is the consensus sequence that represents an average for all mutations, weighted to reflect their individual frequency' [*Eigen, 1993 p.45*].

Clearly, well-defined viral quasispecies will group, or cluster, under most classification schemes. Such a definition seems to be a far more useful approach for classifying computer viruses.

The second and final section of this paper will be published in the October edition of *Virus Bulletin*. It will be an exploration of these issues using the Stoned virus; an explanation of the Data Set and the methods used, and a schematic diagram of the *CARO* classification of Stoned.

Bibliography

Bontchev, 1992: Bontchev, V (1992). The Bulgarian and Soviet Virus Factories. Bulgarian Academy of Sciences.

CARO, 1991: CARO (1991). A New Virus-Naming Convention. Computer Antivirus Research Organization.

Doolittle, 1990: Doolittle, R F (Ed) (1990). Molecular Evolution: Computer Analysis of Protein and Nucleic Acid.

Eigen, 1993: Eigen, M (1993). Virus Quasispecies. Scientific American (July 1993, pp.42-49).

Ford, 1994: Ford, R (1994). Viruses for Sale, a Dime a Dozen. Virus Bulletin (June 1994, p.2).

Heywood & Mcneill, 1964: Heywood, V H; Mcneill, J (1964). Phenetic and Phylogenetic Classification (No 6). The Systematics Association, London.

Hull, 1973: Hull, D S (1973). Darwin and His Critics. Cambridge, MA: Harvard University Press.

ICZN, 1964: ICZN (1964). International Code of Zoological Nomenclature (2nd edition). London: International Commission on Zoological Nomenclature.

Japhet 1994: Japhet, G. (1994). CORAL - The File Correlator. Ver 1.02 [program].

Jardine & Simpson (1971): Jardine, N; Sibson, R (1971). Mathematical Taxonomy. London: John Wiley and Sons, Ltd.

Ludwig, 1993: Ludwig, M A (1993). Computer Viruses, Artificial Life and Evolution: American Eagle Publications, Inc.

Ludwig, 1994: Ludwig, M A (1994). The Collection: Outlaws from America's Wild West. Ver 1 [program]. Tucson: American Eagle Publications, Inc.

Ludwig, 1995: Ludwig, M A (1995). Personal communication.

Peer, 1994: Peer, Y V d (1994). TREECON. Ver. 3.0 [*IBM PC* program].

Sequences 183: Sequences (Vol. 183). San Diego: Academic Press, Inc.

Skulason, 1995: Skulason, F (1995). Personal communication.

Spafford & Weeber, 1992: Spafford, E H; Weeber, S A (1992). Software Forensics: Can We Track Code to its Authors? (No CSD-THR 92-010). Purdue University.

Stojakovic-Celustka, 1994: Stojakovic-Celustka, S (1994) ALIVE (April).

PRODUCT REVIEW 1

InocuLAN for NT

Jonathan Burchell

Cheyenne Software is known and respected for backup and anti-virus products for *NetWare* servers. This month we look at a new product from the company, *InocuLAN for Windows NT*. It has three components: a server element (three 1.44 MB diskettes, one licence disk), one client for DOS/*Windows* workstations (three 1.44 MB diskettes) and one for DOS workstations (two 1.44 MB diskettes).

Documentation

An Administrator guide and a Client guide were included, both of which are substantial, and extremely professionally and attractively produced. In addition to being operational guides, the manuals cover some basic background information on virus symptoms and network protection, and include an appendix covering the more common viruses.

No on-line virus reference is included, but the company has licensed VBASE, an electronic encyclopædia, from *Norman Data Defense Systems*. It is available free of charge to registered users. A list of detected viruses is available within the software.

Server Element

Installation of the server element requires a 486 or higher computer, 16MB or more of RAM, 5MB disk space and *Windows NT v3.5* or above (workstation or server). The software is installed by running set-up in *Windows NT*. The licence diskette must be inserted on installation. It is not required again, but may be used on a re-install: I suspect it is separate only to ease manufacturing and upgrade issues.

The software has three components: server protection, manager or administration front-end, and alert module. The express set-up option installs all components, whilst custom set-up allows components to be installed individually.

The server component is the element which provides the scanning ability, and is copy-protected via the licence disk. The Administrator or Manager module need not be installed on all servers (they can be administered remotely) and may be installed any number of times, including onto workstations which have no server service installed - this makes for great flexibility in server administration. The Alert component is installed on the nominated message centre.

Networking Concepts

Like many of its *NetWare* counterparts, *InocuLAN* allows several servers to be grouped into logical domains. All servers in a domain must be running *InocuLAN for NT*.

The advantage of grouping servers into domains is two-fold. Scheduled scanning need only be set for the master server, propagating automatically to other servers in the domain, as will scanning service information. Also, the master can be set up as the central message centre, allowing reports and logs to be viewed and administered from a central location.

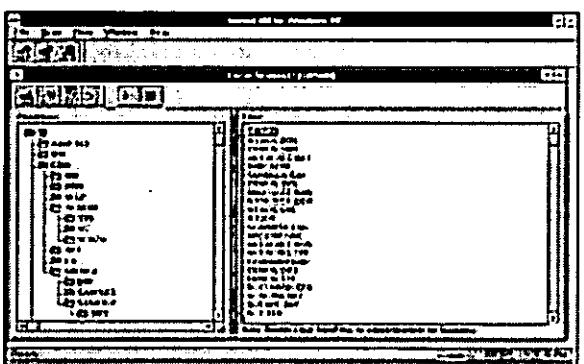
Configuration and administration of all components is accessed via the *InocuLAN for Windows NT* manager icon. The manager consists of three separate modules: the domain manager, the local scanner, and the service manager.

Domain Manager

The domain manager controls and configures domains and scheduled scans. As I had only one *NT* server in my test network, I could not try domain configuration options but, judging from the manual, it is a simple operation to create domains and add servers to, or remove servers from, the domains created.

Domain size may be a single server, or many. A server may be a member of only one domain, and each domain has a nominated master server. As well as domain administration, the domain manger controls scheduled scans, tracking up to 2000 (or 1000 simultaneous) scan jobs. For each, the following information is recorded:

- target drives and directories to scan (a scheduled scan cannot include removable media or mapped drives)
- whether to scan sub-directories of the targets specified
- the CPU usage level (a number from 1-10) at which the background scan is to run
- a list of directories and files to be excluded from the scan (if these do not exist on a particular member of the domain, they will simply be ignored)
- a date and time to start scanning (a repeat interval specified in terms of months, days, hours and minutes)



The local scanner, which controls local and immediate scans, can also access removable media drives and mapped drives.

All files, or executables only, may be scanned. An executable is defined by file extension: the default list is APP, COM, EXE, DLL, DRV, OVL, OVR, PRG, and SYS (BAT and SCR are notable omissions) - extensions may be added or removed. Action to be taken on virus detection includes:

- report only: no action is taken; a message is sent to the Alert module which deals with it as detailed below
- delete file: deletes the file
- cure file: the manual claims that *InocuLAN* can remove, and thus cure, certain infections. It recommends that, even after a cure, you should delete the file and reinstall the original, an attitude we heartily endorse. This raises a question as to whether this option is of use other than if there is no other solution.
- rename file: the default extension for renamed files is AVB (in the event of a file with this extension already existing, *InocuLAN* automatically synthesises an extension of the type AV0, AV! etc). An option allows the default extension choice to be changed.
- move file: moves an infected file to a specified quarantine directory (the default is InocuLAN\Virus)
- purge file: deletes an infected file and guarantees that it cannot be recovered with recover utilities
- rename and move file combines move/rename options

It is also possible to specify scan type: the options are 'fast', 'secure', and 'reviewer'. 'Fast' checks only the beginning and end of a data file, whilst 'secure' checks the entire file and is consequently a little slower. The manual claims that 'reviewer' detects virus-like activity within a file (a heuristic approach perhaps), whilst the on-disk READ.ME file claims that 'reviewer' uses a database of garbage virus strings.

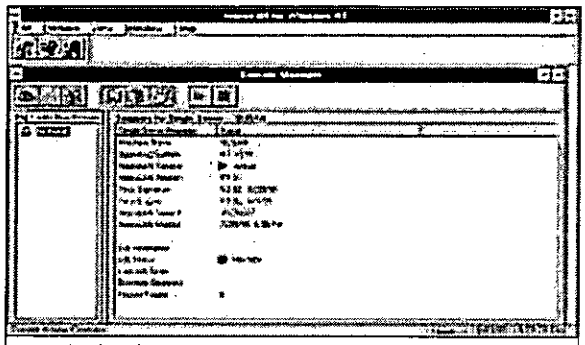
I suspect 'reviewer' contains signatures from test-sets which do not represent true viruses. The *VB* test-set has only genuine, viable infected files. The manual states that using 'reviewer' may cause false positives - I set the scanning to 'reviewer' for the detection tests. Further options in this section allow starting, stopping and rescheduling of jobs.

Local Scanner

The second component of the *InocuLAN* manager is the Local Scanner. This module controls local and immediate scans. Unlike the scheduled scanner, it can access removable media drives and mapped drives.

Options for the scan are broadly similar to those outlined for the scheduled scan, with the exception of job start and repeat information. Additionally, it is possible to request that *InocuLAN* prompts the user before taking any action on an infected file, and that it 'beeps' the workstation speaker when an infected file is discovered.

Selecting what to scan is specified via a graphical tree representation of the drive, which makes it extremely easy to indicate specific directories and files to be included in or excluded from the scan. Unfortunately, I could see no way of saving the choices for another session, or indeed of keeping a list of different types of immediate scan jobs.



The domain manager controls and configures domains and scheduled scans, tracking up to 2000 scan jobs.

Service Manager

The final option in the manager module allows for starting and stopping of the scanning service. This sets whether scanning service starts automatically when a *Windows NT* machine is booted, and sets various parameters affecting how often the service manager should scan job queues, poll apparently dead servers, and hold finished jobs in the queue.

It is also possible, with the event and the scan logs, to set how many messages to retain in the log file (this may be set between 10 and 1,000), after how many days to purge records automatically, and the level of information to be stored. This can be any combination of critical, warning and informational messages. The event and scan logs are accessed directly from within the relevant program sections.

The included *Windows* help files are informative, attractive and easy to use. They offer a dual pane mode, with contents in one screen and the selected entry in another, making it quite simple to 'read' the manual on-line.

Alert

Whenever an *InocuLAN* server or workstation client produces an event (such as detecting a virus), it sends a message to the server nominated as the domain master. There, it is intercepted by the Alert module, processed, and added to the central 'master' database of alerts. A received alert may cause any of the following actions to take place:

- a broadcast message sent to nominated users or groups
- a pager message (numeric or alpha-numeric) sent to a nominated group of recipients. Requires a modem connected to a server machine to access pager service. The message sent consists of a detection code number, a machine ID number and a user-defined custom code.
- SNMP trap messages sent across the network to an SNMP management product such as *NetWare* Management System (NMS) or HP OpenView. Either IPX or TCP/IP may be selected as the transport mechanism.

- Trouble Ticket: this option allows a list of printers to be defined. *InocuLAN* will print a Trouble Ticket automatically when an alert is received.
- Email: this option, which requires *Microsoft Mail*, allows a nominated list of recipients to be notified of alerts via email. It is possible to specify the 'To:, CC: and Subject:' parts of the header as well as to attach a specified list of files to the message (for example, the event log).

The eagle-eyed among you will have spotted that none of the options discussed so far control real-time checking of file read and write. Unfortunately, the interface to the *Windows NT* file system is such a closely guarded secret that no anti-virus vendor has been able to provide real-time file checking for *NT* server products. *Cheyenne* is no exception.

Thus, there can be no real-time protection for server-to-server or workstation-to-server transactions when both systems are using *Windows NT*. As, at the moment, there are no *Windows NT*-specific viruses, this may not be much of an issue. DOS sessions within an *NT* (or *OS/2*) workstation, or on a DOS or DOS/*Windows* platform, may be protected by loading appropriate *InocuLAN* client software.

The critical component is Immune, a TSR which provides real-time checking of files as they are accessed in a DOS session or on a DOS/*Windows* workstation. Immune can send alerts across the network to the Alert master, providing for centralised monitoring of real-time workstation activity.

The Immune/Server communication relies on IPX packets being available as a transport mechanism, which is rather a shame, as many *Windows 95/NT* networks will be NetBEUI or TCP/IP only. However, *Cheyenne* intends to provide support for TCT/IP in the next release of the product.

Results

The main problem with the virus detection provided by the main scanner seems to be the lack of identification of the SMEG and Cruncher polymorphics (plus a slight wobble on some of the MtE variants) and that some basic signature data for the 'Standard' and the 'In the Wild' test-sets is missed. Having said that, however, the detection ratios show the kind of performance which could easily be tuned to 100%.

As is shown in the results table, there are obvious problems with real-time detection. This aspect, represented by the Immune detection figures, is not good enough to guarantee a good level of viral immunity. I suspect that this lower figure comes from the twin pressures of maintaining two code bases and keeping the TSR element for DOS to a reasonable size. *Cheyenne* will shortly be providing VxDs for *Windows* and *Windows 95*, and a similar system for *Windows NT*.

Conclusions

InocuLAN for NT brings the sophistication of big league *NetWare* products to *Windows NT*. It has a user interface which makes the most of the *Windows* Graphical User Interface, and helps ease administration of large networks. The inclusion of features such as domain administration, and sophisticated alert and messaging systems, set it above *SWEEP for NT* in terms of features and may make it more suitable for large sites.

I do have a few gripes, however. The concept of domains, scheduled scans and local scans in the Manager module is a little confused. In a large network, I might also want Alert to function across multiple domains, rather than having to set it up for separate domains.

It also seems surprising that the signature database cannot be automatically propagated to all domain members (or to all members of the visible network). This feature is planned for future release, according to *Cheyenne*.

Having said that, the features and quality of this package are astounding, even more so when combined with the knowledge that this is the first version. Detection ratios, except for some problems with the polymorphics, are good (see results, below), though not as good as those for *SWEEP for NT*.

The good news is that *Cheyenne* feels it will crack the problems of real-time checking on the server. Once this has been achieved, the high detection rates, together with the superb user interface and server administration, mean that this will be a product to consider in any installation for *Windows NT*.

InocuLAN for NT

Detection Results

Main Scanner:		
Standard Test-Set[1]	229/230	99.6%
In the Wild Test-Set[2]	120/126	95.2%
Polymorphic Test-Set[3]	3732/4796	77.8%
Immune:		
Standard Test-Set[1]	228/230	99.1%
In the Wild Test-Set[2]	118/126	93.7%
Polymorphic Test-Set[3]	1214/4796	25.3%

Technical Details

Product: *InocuLAN for NT.*

Developer: *Cheyenne Software Inc*, 3 Expressway Plaza, Roslyn Heights, NY 11577 USA. Tel +1 516 484 5110, fax +1 516 629 1853, email cheyenne@cheyenne.com.

Price: US$895 (1 server), US$3995 (5 servers), including upgrades (every two months), and licences for all DOS, *Windows*, and *Macintosh* machines connected to the server(s).

Hardware used: Client machine - 33 MHz 486, 200 Mbyte IDE drive, 16 Mbytes RAM. File server - 33 MHz 486, EISA bus, 32-bit caching disk controller, *NetWare 3.11*, 16 Mbytes RAM.

Each test-set contains genuine infections (in both COM and EXE format where appropriate). For details of the Standard test-set, see *VB*, January 1994, p.19 (file infectors only). For details of In the Wild and Polymorphic test-sets, see *VB*, August 1995 p.19.

PRODUCT REVIEW 2

IBM AntiVirus

Dr Keith Jackson

IBM AntiVirus has been reviewed by *VB* several times before: version 1 for DOS in January 1993, the *OS/2* version in August 1993, as part of *PC-DOS* in January 1994, and the *NetWare* version in February 1995.

This review is of version 2.2, which can be used with DOS, *Windows* or *OS/2*. It was provided for review on three 3.5-inch, low density floppy disks. *IBM* claims that its anti-virus software 'is the software that IBM uses to protect its own personal computers' [*I should hope so! Ed.*], and that it is 'designed to detect and remove viruses from your system as simply and reliably as possible'.

Documentation

The documentation took the form of an A4 ring binder, containing 101 pages about its DOS and *Windows* versions. I have no real complaints about the manual - it is readable, well-indexed, explains the basics well; however, it does lack some explanation of fine details, such as possible errors.

The on-line documentation contains a list of 3636 viruses which *IBM AntiVirus* claims to be able to detect. Another thousand lines of cross-reference information are provided, which permit searching for virus name through a common alias. Also included is a more detailed explanation of 153 of the more common viruses, a set which seems well chosen. Along with details of the Family/Classification of each, a paragraph explaining how the virus operates is provided.

Installation

Two different methods of installation are described in the documentation, one of which operates under DOS, one requiring *Windows*. Curiously, both methods install the files required for operating the product under *Windows*.

Installation had to be done using DOS, as the *Windows* SETUP seemed to be missing from the master disks - a bad omen? Shortly after installation commenced, the program asked whether an 'Emergency Diskette' should be made. Being cautious, I answered yes. It proved impossible: the program requested that disk 3 was inserted, then failed to recognise it correctly. I restarted, and re-installed without making an emergency diskette. No matter what I did, the program stopped after installing 29 files (750 KB), produced the wonderfully vague error message: 'Error in transferring IBM AntiVirus files', and refused to continue.

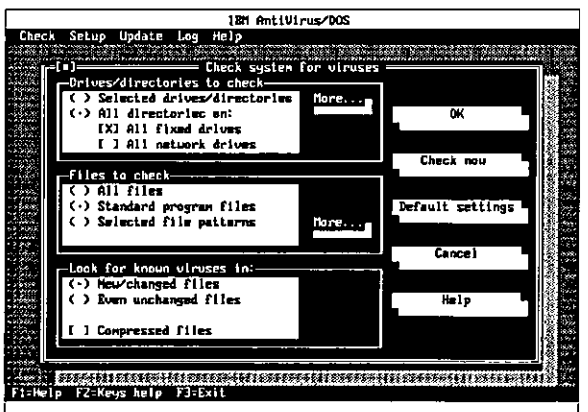
This error was at least consistent - another set of disks sent to *VB* at the same time exhibited the same problem. So here I am, for the second consecutive month with a product which would not install correctly. After a few tests, and several phone calls to the manufacturer, it was apparent that the second and third disks in the set (which seemed identical) contained files dated 1987 (I get all the most recent stuff!): they referred to mouse drivers with instructions provided in Swedish, French, German (and seemingly every conceivable European language).

The fact that the second and third disks were identical, and contained nonsense, was not the source of the problems described above. The installation process did not even get as far as asking for disk 2 before it died.

To cut a long story short, I downloaded a new version of the software from the *IBM* BBS. This worked properly, and installed under DOS and *Windows*. A plaudit is in order here for the Technical Support people, who did well in digging me out of my hole. I always received sensible advice, phone calls were returned promptly, and a solution did eventually appear. Maybe they've had a lot of practice! Just a joke...

The new downloaded version of the product gave no trouble. The DOS version installed 49 files which occupied 1.65 MB; the *Windows* version, 57 files in 2.99 MB. DOS installation takes significantly less time than that for *Windows*. *IBM AntiVirus* installed all its files into a personally selected location, and is able to alter AUTOEXEC.BAT, or store the desired changes in a separate file for later manual insertion.

Under both DOS and *Windows*, the install program offered to make an emergency diskette containing a stripped-down *IBM AntiVirus*, for use *in extremis*. I am sure that many users would infer from its name that the emergency diskette would facilitate resurrection of a PC if anything went wrong, i.e. that the floppy was more than a diskette-based virus detection system - which it is not.



Various options are available for scanning. The time taken by *IBM AntiVirus* to complete a scan, after its initial execution, compares favourably with the market leaders.

De-installation is very simple, albeit not self-evident. If the *Windows* version has been used, it is simply a matter of removing a line from the file WIN.INI, removing two lines from AUTOEXEC.BAT, and manually deleting the *Windows* group and the associated *IBM AntiVirus* icon.

Disk Checking

The first time the product executes, it says it is 'initialising its database', i.e. it searches through all hard disks to decide which files should be checked, scans each, and, if uninfected, calculates a checksum for each. This takes a long time (11 minutes 2 seconds under *Windows*, 9 minutes 59 seconds with DOS), but only happens on installation. All subsequent executions use this database to verify that files are unchanged, and scanning is then required only if something is found to be new, or altered in any way.

Many setup options are provided: an automated check (each boot, daily, weekly or monthly), checking inside compressed files (this is switched off by default, and adds considerably to the overall time taken to check a disk), scanning of high as well as 'normal' memory, and specification of any desired combinations of drives/files.

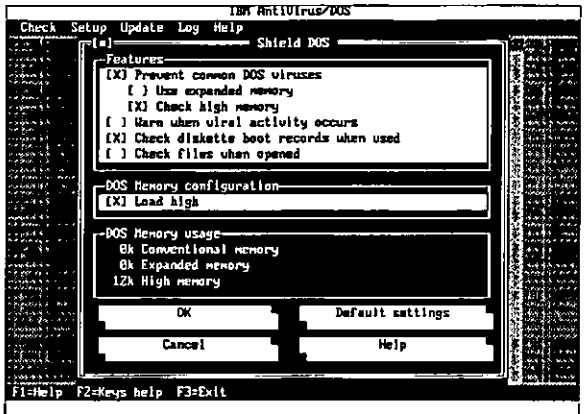
Although all options were left at their default values, the DOS version of the product detected 656 objects which required scanning, but the *Windows* version only found 648. *IBM* states that this is due to the fact that *Windows* locks certain files so they cannot be scanned. In both cases, 35 seconds was spent scanning memory and counting how many objects should be scanned (mainly the latter) every time the hard disk of my test PC was checked.

Subsequent executions of the product were much faster than the initial one. The *Windows* version checked my test PC's hard disk in 1 minute 40 seconds, when scanning for new or unchanged files. Under DOS, this took 1 minute 12 seconds. Using the 'scan unchanged files' option, the time taken rose to 7 minutes 20 seconds. This confirms the speed-up offered by the tactic of looking to see which files have changed, and scanning only those which have altered. In comparison, *Dr Solomon's AVTK* performed the same scan in 1 minute 39 seconds, and *Sophos' SWEEP* in 1 minute 34 seconds.

Accuracy

The samples used for testing are listed in the Technical Details. Of the 239 parasitic viruses, 38 were detected as definite infections, 197 as probable. Only four parasitic viruses (WinVirus_14, 8888, and two copies of Starship) went undetected. All nine boot sector viruses were detected correctly, giving an overall detection rate of 98.3%. All 500 Mutation Engine (MtE) samples were detected correctly.

Results in all sets were identical whether the DOS or the *Windows* version was used. When a ZIP file containing many MtE test samples was checked, *IBM AntiVirus* said only that the ZIP file was infected, and gave no indication of how many infected files were present.



The memory-resident component of *IBM AntiVirus*, DOS Shield, contains features allowing for various types of checking. The option shown above is active by default.

The viruses found by this product are split into 'definite' and 'probable' infections. The majority, 85%, are detected as 'probable', though they are viruses. The false positive rate was zero. As for Number_of_the_Beast, Vacsina and Yankee, some samples were detected as 'definite'; others, only 'probable'. Why? *IBM's* answer is that the product only identifies a virus as 'definite' if it is byte-for-byte identical with the one analysed; if similar, it is described as 'probable'.

Memory-resident Program

IBM AntiVirus includes a memory-resident feature called DOS Shield, comprising several components which are loaded sequentially, as desired. The separate parts claim to 'Prevent common DOS viruses', 'Warn when viral activity occurs', 'Check diskette boot records', and 'Check files when opened'. Each component provides a concise onscreen explanation of its function when it loads into memory. Only the first and third of these components are active by default; the others must be explicitly selected.

The setup screen gives an accurate indication of how much memory various combinations of these components will use. Although high memory can be used to reduce the amount of conventional RAM that is required, only one component (Prevent common DOS viruses) can use expanded memory. Use of high memory and/or expanded memory can be altered at will by the user.

When all four components are active simultaneously, 18 KB of conventional (high) memory, and 16 KB of expanded memory is required; an eminently acceptable total.

Memory-resident software is notoriously difficult to test with accuracy, but I did my best. With all the memory-resident components active, I used *Norton Commander* to copy a test-set containing one of each of the viruses listed in the Technical Details section (148 viruses in total) from one disk to another. DOS Shield reported 28 files as infected - not encouraging. *IBM's* rationale is that DOS Shield should focus on those viruses which the user is likely to encounter.

I often use 4DOS (a command interpreter which is a replacement for COMMAND.COM): when this was in use, and infected files were copied using the COPY command, all infected files were detected correctly.

After 52 files had been copied, this command produced the onscreen error message 'Too many open files' for each file it attempted to copy. After this, the PC produced that same error in response to every DOS command, and a reboot was required. If COMMAND.COM was used, COPY terminated when the first virus-infected file was encountered, and an error message appeared on screen. Version 2.3 of the product, according to *IBM*, does not contain this problem.

The memory-resident program did not detect virus-infected files within a compressed ZIP file. This is unsurprising, as such a facility would probably add a large overhead to system execution. However, when I extracted virus-infected files from a compressed ZIP file, there was no complaint from the software. Given that this created many new virus-infected files, it did seem something of an omission.

I tested the overhead added by the memory-resident software by copying 20 files (585 KB): the time taken to do this was approximately the same whether or not DOS Shield was installed, and no matter which component parts were active. Oddly, my timing measurements showed much greater variation when DOS Shield was installed. Given that the variation could be anything up to a one-second alteration in an 11 second file copying time, this was much larger than any possible measurement error which I might have made. I cannot think of any reason why this should happen.

The documentation does not explain the constraints imposed by the behaviour blocker component (it never does!). Therefore I formatted a floppy disk, ran SYS, then ran *Norton's* formatting program, and even edited absolute sectors of a floppy disk. All to no avail - I could not induce an error message. Contact with *IBM* revealed that the company has designed DOS Shield to be able to distinguish between viral and normal system activity.

The Rest

Although DOS and *Windows* versions of *IBM AntiVirus* were provided, I could detect no difference between the two, apart from some screen representation details. Even the selections available on the drop-down menus are almost identical. On my test PC it took 10.9 seconds for the DOS version of *IBM AntiVirus* to load. Given that this was a 33 MHz 486, it is likely that loading could become turgid on a slow 386, and unusable on anything less powerful.

Disinfection facilities are provided with *IBM AntiVirus*, but in common with my usual practice, I have not assessed this capability. Be safe, delete all infected files; you know it makes sense. *IBM AntiVirus* maintains three logs files whilst disks are being checked: these provide thorough details of what happened on the last execution, the previous execution, and a cumulative log of all previous checks.

Conclusions

Given the problems I had with the version of *IBM AntiVirus* originally provided for review, the phrases 'thorough testing' and 'lack of' (in no particular order) spring to mind. If *IBM* cannot come up with software which works when they know it is being provided for a review, what chance do ordinary punters have?

IBM AntiVirus detects viruses accurately and in a timely fashion. By combining the features of a scanner and a checksummer, the time taken to perform the initial check of a hard disk is quite slow. However, this only happens once, and all consequent checks are carried out more quickly than would be the case if scanning alone were used.

Indeed, using its tactic of combining a scanner and a checksummer, *IBM AntiVirus* can check disks at speeds which are faster than most anti-virus programs. Scanners which blindly search rarely-accessed corners of a hard disk are blundering through their search process for no reason, so it does seem logical to try and combine scanning and checksumming. As long as it is done carefully.

The memory-resident component is not very good at spotting virus-infected files, and does not seem to prevent a user carrying out harmful actions. However, it occupies very little memory, and does not impose a large overhead. I suppose we should be grateful for small mercies.

All this takes me back to the comparative scanner review published in the July edition of *VB*. This contained the conclusion that *IBM AntiVirus* was 'one of the slowest products tested'. I disagree. The above review has shown that this is only true the first time a disk check is invoked. On subsequent checks, *IBM AntiVirus's* combination of a scanner and a checksummer makes it faster than most products which rely solely on scanning.

Technical Details

Product: *IBM AntiVirus v2.2* (no serial number available).

Developer/Vendor (UK): *IBM UK*, Normandy House, Alencon Link, Basingstoke, Hants, RG21 1EJ. Tel 01256 314558, fax 01256 332319.

Developer/Vendor (USA): *IBM Corporation*, Long Meadow Road, Sterling Forest, NY 10979-0700. Tel +1 914 759 2901, fax +1 914 784 6054. Note also that *IBM* provides support for its *AntiVirus* program through its usual outlets in almost every country in the world. The documentation contains a voluminous list of contact addresses and telephone numbers.

Availability: Any *IBM PC*, PS/2, or 100% compatible with 640 Kbytes of RAM, and OOS version 3.3 or above.

Price: 1-250 users, £1000; 251-500, £2000; 501-1000, £4000; 1001-2000, £6500; 2001-3000, £9500; 3001-5000, £12,500; 5000+ on application only. Includes quarterly updates.

Hardware used: A 33 MHz 486 PC clone with 3.5-inch (1.44 MB) floppy disk drive, 5.25-inch (1.2 MB) floppy disk drive, a 120 MB hard disk and 4 MB of RAM, using *MS-DOS v5.00*, *Windows v3.1* and *Stacker v2*.

NB: For full details of viruses used for testing purposes, please see *VB*, May 1995, p.23.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
CompuServe address: 100070,1340

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield, CT 06877, USA

Tel +1 203 431 8720, fax +1 203 431 8165

END NOTES AND NEWS

***Compsec 95* will take place in London** from 25-27 October 1995. For details on the conference, contact Sharron Emsley at *Elsevier Advanced Technology* on Tel +44 1865 843721, fax +44 1865 843958, email s.emsley@elsevier.co.uk.

Information Security on the Internet is a **two day conference** taking place at the *Cumberland Hotel* (London, UK), on 25/26 September 1995, with post-conference workshops on 27 September. Tel +44 181 332 1112, fax +44 181 332 1191 for information.

The ***22nd Annual Computer Security Conference and Exhibition*** will be held in Washington, DC from 6-8 November 1995, under the auspices of the *Computer Security Institute* (*CSI*). The conference will feature over 120 sessions on various topics. Further information is available from the *CSI* on Tel +1 415 905 2626, fax +1 415 905 2626.

The **next round of anti-virus workshops being held by *Sophos Plc*** is scheduled for 22/23 November 1995. The two-day seminar will take place at the company's training suite in Abingdon, and costs £595 for both days (or £325 for one day only). The first day's sessions comprise an introductory course on computer viruses, and the second day is an advanced virus workshop. More information is available from Julia Line on Tel +44 1235 544028.

A **new *Macintosh* virus** has been found in the wild: HC-9507 causes unusual system behaviour, linked to the day of the week and the time: screen fade-in/fade-out, automatic entering of the word 'pickle', or system shutdown/lockup. It infects HyperCard stacks under *Apple Macs* running system 6 and 7.

IBM has announced the release of **an integrated suite of anti-virus products and services**, including software which protects PC users by detecting and removing more than 6000 strains of computer virus. The *Desktop Edition*, targeted at home users and small businesses, runs on *OS/2*, DOS, and *Windows*, with *Windows NT* and *Windows 95* support planned for late 1995. Aimed at large businesses and client/server environment, the *Enterprise Edition* includes *IBM AntiVirus for OS/2*, OOS, *Windows*, and *NetWare*. For information, contact Andrea R. Minoff at *IBM*; Tel +1 914 759 4713, email minoff@watson.ibm.com.

The ***European Security Forum Annual Congress*** will be held in Cannes, France, from 15-17 October 1995. Information on the conference can be obtained from June Chambers at the European Security Forum's London offices; Tel +44 171 213 2867, fax +44 171 213 4813.

Fischer International is about to launch a **data security product for *OS/2*, *Watchdog***. The current product line provides security for DOS and *Windows*. *Watchdog for OS/2* is now undergoing beta-testing, and will start shipping when *IBM* releases its new security hooks for *OS/2*. Further information is available from Liz Menches at *Fischer*; Tel +44 1923 859119, fax +44 1923 859151.

S&S International will be holding two rounds of ***Live Virus Workshops***; on 18/19 September and on 9/10 October 995. Cost for the two-day seminar is £680 + VAT. Further details can be obtained from *S&S International*; Tel +44 1296 318700, fax +44 1296 318777.

The *National Computer Security Association* (*NCSA*) has organised a ***Firewall Product Developers' Consortium*** (*FWPD*) to bring together the major vendors of network and Internet firewall products. According to Dr Peter Tippett, *NCSA* president, the effort is meant 'to bring together the vendors of firewall products, consumers who buy these products, and the best security experts we know'. Information on the initiative is available from Bob Bales at the *NCSA*: Tel +1 717 258 1816, fax +1 717 243 8642, email bbales@ncsa.com.

ISSN 0956-9979 AUGUST 1996

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Hold the front page!** Just when you thought it was safe, the macros are back. This month sees the appearance of Laroux, the first *Excel* virus, in the wild. Turn to p.9 for the low-down.

• **Ethics et cetera.** Distribution of viruses has always been a thorny problem, and the ever-increasing growth in the use of the Internet makes this problem more real than ever. Sarah Gordon presents some of her findings and thoughts on p.14.

• **Praying for salvation...** Hare Krishna is not a phrase which one would usually associate with computer viruses: this is a fact which may now have to change. Hare.7610 is a new virus, laden with interesting features... and it's in the wild. See p.11 for an analysis.

CONTENTS

EDITORIAL
What a Wonderful World 2

VIRUS PREVALENCE TABLE 3

NEWS
1. *MS* Licenses AV... Again 3
2. Secure Checking? 3

IBM PC VIRUSES (UPDATE) 4

FEATURE 1
Generic Decryption Scanners: The Problems 6

VIRUS ANALYSES
1. Excel Yourself! 9
2. Hare Krsna: ISKCON too Far! 11

FEATURE 2
Viruses on the Internet 14

PRODUCT REVIEWS
1. *CPAV for NetWare* 18
2. Survival of the Fittest? 21

END NOTES & NEWS 24

EDITORIAL

What a Wonderful World

Every so often in this business, I get a strange feeling. It always follows a thought of the 'what if...?' type. Some technical details will follow, sometimes of a viral technique, often of flaws in any one of more than thirty anti-virus products, possibly even of a vulnerability in UNIX or *Windows NT*. After a few minutes of mental elaboration, that strange feeling dawns: terrible inevitability.

Take, for example, macro viruses. In August last year, along came Concept. Shortly after this, many possibilities occurred to those involved, most notably the likelihood of other applications being affected by similar creations in the future. One word kept recurring... inevitable.

"after a certain distribution of a given virus has occurred, it will be extremely difficult to eliminate that virus from the wild"

From the massed ranks of these 'other applications', one primary contender emerged: another member of the vastly-popular *Microsoft Office* suite, the most over-featured spreadsheet yet to hit the market (presumably only to be overshadowed by subsequent versions...), *Excel*. Just as *Word* is the common denominator of word processing systems, *Excel* is fast becoming the spreadsheet format of choice. As if that wasn't enough, it incorporates a macro-ing system (*Visual Basic for Applications*, or *VBA*) orders of magnitude more powerful than *Word's* (*WordBasic*). That word again: inevitable.

One year later, it has come to pass. Like an echo of Concept, Laroux raises its head above the parapet of conjecture and into the blinding light of reality; the first (well, the first to be *discovered*) *Excel* virus (see analysis, p.9).

The similarities with Concept are marked. There is no payload. The code contains some curiosities, but it is starkly functional, and by and large works well. It began to circulate in the wild before anyone noticed it, and although at this point the size of the distribution is not known, the fact that it is so firmly in the wild gives it a significant advantage. Not only is it the first of its type, it has also been placed (presumably intentionally, although conceivably by accident) into the wild by its author. We also don't know how long Laroux has been in the wild, which will have at least some bearing on how far it has spread – another significant factor determining the eventual spread of the virus is where it was introduced. Clearly, initially introducing the virus into a large multinational company with thousands of *Excel* users worldwide should result in a much wider spread than uploading one spreadsheet labelled 'Interesting numbers' to a bulletin board in Venezuela.

There is, alas, only one clear distinction here – that between a virus being in the wild, and being only found in laboratories. Once it is out there, the question of how much it is out there is secondary in importance.

It is to be expected that things will now follow the path established by Concept – we will see a number of rushed, and correspondingly careless, copycat attempts, and quick and dirty modifications by other authors; then we will see slicker, better-written follow-ups. However, by this time, anti-virus companies will have rushed around and come up with a range of 'fixes' for the problem. One hopes that they will be able to make faster headway than was possible with the Concept virus, as the basis of the file format of *Excel* spreadsheets is the same OLE format used for *Word* documents, and most major manufacturers have already built parsers for this into their scanners. However, there is still plenty of complexity to go around, as *Excel* has its own data formats hiding underneath the OLE structure. For this it's back to reverse engineering, as *Microsoft* is bound to be as recalcitrant as ever with its information.

If the *Excel*-using community is lucky, defences will be developed before the virus is able to gain the firm foothold in the real world that Concept has managed. The idea of 'critical mass' has a place: after a certain distribution of a given virus has occurred, it will be extremely difficult to eliminate that virus from the wild (by wiping it out amongst the user community), regardless of the effectiveness of the defences introduced, in much the same way as stopping a nuclear reaction becomes next to impossible once the neutron flood is too strong. Perhaps it's not such a wonderful world after all...

NEWS

MS Licenses AV... Again

At the beginning of July, *McAfee Associates* announced that it has licensed 'portions of its anti-virus technology' to *Microsoft Corporation* for use in *Microsoft's* Internet software products.

As readers who follow Internet trends will be aware, *Microsoft* is involved in a battle with *Netscape* for domination of the WWW browser market – *Microsoft's Internet Explorer* and *Netscape's Navigator* have been engaged in a ferocious 'features war' for several months now.

Already, anti-virus add-ins to *Navigator* are available (though see End Notes and News for further details). This move by *Microsoft* appears to be designed to remove *Netscape's* advantage in this area. It remains to be seen how *Microsoft* will provide virus information updates – *VB* is mindful of the long drawn-out and thoroughly painful *MSAV* fiasco ▮

Secure Checking?

At the end of June, *Secure Computing* (formerly *Virus News International*) announced the creation of the *Secure Computing Checkmark* scheme. Labelled as 'security product certification', it will apply initially only to anti-virus products, although *Secure Computing* intends to extend the scheme to encompass other security products 'in due course'.

The testing revolves around detecting those viruses in the wild (Joe Wells' *WildList* will be used as the primary source for information as to which viruses are out there). A product must score 100% in the tests to be awarded the *Checkmark*. Once the *Checkmark* has been obtained, the manufacturer is granted the right to use a logo on its marketing material. Additionally, a certificate is issued for the *CheckMorked* product, stating that it has been approved by *Secure Computing*: this may be included in product packaging.

The scheme is superficially similar in nature to the *NCSA* system, but will not, one hopes, be beset by problems to the same extent. The costs are difficult to calculate, but a press release from *Secure Computing* places the cost between £2200 and £9500 per product for the first year, and £1800 and £4500 per product in subsequent years.

Developers already signed up (initial testing for which will take place later this year) include *Command Software*, *DataFellows*, *ESaSS*, *Reflex Magnetics*, *S&S* and *Symantec* ▮

Corrections: In the July 1996 scanner comparative review, *VB* incorrectly listed *PCVP's* version number as 2.23: it should have been 2.33. For the same product, infected floppy scan time was listed as 31, but should have read 0:31; i.e. 31 seconds.

Further, Gregg from *Command Software* points out that FDISK /MBR will not remove Boot.437 from a hard drive – SYS C: is required (where C: is the drive in question).

Prevalence Table – June 1996

Virus	Type	Incidents	Reports
Concept	Macro	67	19.5%
Form.A	Boot	34	9.9%
Parity_Boot	Boot	28	8.1%
AntiEXE	Boot	25	7.3%
NYB	Boot	17	4.9%
Junkie	Boot	14	4.1%
AntiCMOS.A	Boot	12	3.5%
Ripper	Boot	11	3.2%
Empire.Monkey.B	Boot	9	2.6%
Sampo	Boot	9	2.6%
Quandary	Boot	8	2.3%
Imposter	Macro	5	1.5%
Jumper.B	Boot	5	1.5%
Stealth_Boot.C	Boot	5	1.5%
Telefonica	Multi	5	1.5%
Burglar.1150	File	4	1.2%
Bye	Boot	4	1.2%
Empire.Monkey.A	Boot	4	1.2%
Natas.4744	Multi	4	1.2%
Stoned.Angelina	Boot	4	1.2%
WelcomB	Boot	4	1.2%
AntiCMOS.B	Boot	3	0.9%
Feint	Boot	3	0.9%
Manzon	File	3	0.9%
Russian_Flag	File	3	0.9%
She_Has	Boot	3	0.9%
Stat	Boot	3	0.9%
Stoned.Stonehenge	Boot	3	0.9%
V-Sign	Boot	3	0.9%
Barrotes	File	2	0.6%
DieHard	File	2	0.6%
EXEBug	Boot	2	0.6%
Stoned.NoInt	Boot	2	0.6%
TaiPan.438	File	2	0.6%
Tentacle	File	2	0.6%
Wazzu	Macro	2	0.6%
Other [1]		28	8.1%
Total		344	100%

[1] The Prevalence Table includes one report of each of the following: Amoeba, Boot.437, BootEXE.451, Bug70, Cascade, Crazy_Boot, Cruel, Diablo, DMV, Fat_Avenger, Hidenowt.1747, Int40, J&M, Lozinsky.1958, Mongolian, Naughty, Neuroquila, Nomenklatura, One_Half.3544, Screaming_Fist.696, Stealth_Boot.E, Stoned.LZR, Stoned.Michelangelo, Stoned.Spirit, Trojector.1463, Unashamed, Vacsina, WBoot.

IBM PC VIRUSES (UPDATE)

The following is a list of updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 21 July 1996. Each entry consists of the virus name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or a dedicated scanner which contains a user-updatable pattern library.

Type Codes

C	Infects COM files	M	Infects Master Boot Sector (Track 0, Head 0, Sector 1)
D	Infects DOS Boot Sector (logical sector 0 on disk)	N	Not memory-resident
E	Infects EXE files	P	Companion virus
L	Link virus	R	Memory-resident after infection

Alho.676

CN: An appending, 676-byte, fast, direct infector. It contains the text: 'CTRL,SHIFT & ALT keys are reserved for internal use' and, at the end of infected files, the string 'Alho'. The virus contains an internal counter: after 20 generations it hooks the interrupt Int 09h and monitors a usage of Ctrl, Shift or Alt keys.

```
Alho.676        8A24 2688 25F3 A406 1F33 D2B8 0925 CD21 C350 1E33 C08E D8F6
```

AOS.863

CER: A stealth, encrypted 863-byte virus which disables VSAFE (the memory-resident component of *Microsoft Anti-Virus*). The virus contains the text: 'M*A*D*#*C*O*W*#*D*I*S*E*A*S*E', and all infected files have their time-stamps set to 6 seconds.

```
AOS.863         5059 BA01 FAB8 4559 92CD 1692 9292 9292 9292 B9AF 01BB ????
```

Blin.1488

ER: An encrypted, polymorphic, 1488-byte virus containing the text: '[Treblinka V 2.01 by Blas Pascal] . Argentina . xx/06/1995 .' All infected files have the string 'BP' located at offset 12h (checksum in EXE header). The following template can be used to detect the virus in memory.

```
Blin.1488       3DCA B075 036B F8CF 3D00 4B74 052E FF2E 4F00 E884 0406 9C60
```

Caco.2965

CER: A stealth, appending, 2965-byte virus containing the plain-text message: 'CACO VIRUS GENE-101. COCO, ALDO, CHINO, OTTO. DOOM-TEAM &CREADORES DE VIRUS&'. All infected files have their time-stamps set to 60 seconds.

```
Caco.2965       33DB B803 FECD 215E 33D2 5681 FB45 4675 232E 3AAC 920B 771F
```

Caco.3310

CER: A stealth, appending, 3310-byte variant of the above virus. All infected files have their time-stamps set to 60 seconds.

```
Caco.3310       33DB B8FF FDCD 215E 33D2 5681 FB41 4C75 232E 3AAC EB0C 771F
```

Critter.1015

CR: An appending, 1015-byte virus containing the text: '[PGa] a critter from DC has infected U ;)' which is visible at the end of all infected files. The virus reinfects already-infected programs.

```
Critter.1015    BA34 1280 FC30 B430 7420 CD21 81FA 1234 B8?? ????  0E8B D881
```

Deadwin.1228

CER: An encrypted, appending, 1228-byte virus containing the text: 'Dead to Windows!' and 'hard disk destroyed!'. The virus payload (triggering on 13 November, 21 June and any Friday) includes the formatting of disks and screen effects.

```
Deadwin.1228    B948 022E 8B3C F7D7 23FD F7D5 2E21 2C2B 093C F7D5 4646 E2EB
```

Delta.1163

CER: A stealth, encrypted, appending, 1163-byte virus. It contains the text: 'Good bytes from (DEL)ta Virus !!! Reset in 30 seconds !' and 'Brazil - 02/96'. This virus triggers on 4 November: its payload changes the CMOS data, disabling the hard disk and destroying the information on floppy drive types. The virus then reboots the system.

```
Delta.1163      1F0B 07BE 2300 03F5 8BFE B980 043E 8A66 04FC AC32 C4AA E2FA
```

Epsilon.513

EPR: A 513-byte (effective virus length) virus which contains the encrypted string: 'CDMMAND' and the plain-text message: '<Epsilon 1.0 (C) 15.3.1995 B.T.Pir8>'. Unlike other companion viruses, it creates COM files that are not marked as hidden and have different lengths (the virus appends to its code a variable number of 'rubbish' bytes).

```
Epsilon.513     3DFC 0C75 04B8 F3F3 CF60 3D00 4B75 03E8 0600 612E FF2E BD02
```

IVP.495

CN: An appending, 495-byte, fast, direct infector. It contains the plain-text message: 'BiATcHSiQB0Y' and 'Hi, my name is Kevin S, and I live in you kompewtor! EyE yEWs 2 bE LeeT, SeW PHeAR mAH! (Fairfax, Va)'. All infected files are marked with the signature 'CA' located at offset 0003h.

```
IVP.495         A5C6 865F 03E9 899E 6003 C786 6203 4341 B905 00E9 0000 5133
```

HLLP.7000

CEN: An appending virus, 7000 bytes long, which contains the text: 'Superviced by Stork Oeba 5/1/95'. Because it was written in high level language, other plain-text messages are visible; e.g. 'Portions

Copyright (c) 1983,90 Borland', 'This program cannot be executed in a Window's shell.', and 'This program requires Windows.'

```
HLLP.7000          0343 4F4D 0345 5845 5589 E5B8 2C04 9A7C 027B 0081 EC2C 04C4
```

IVP.674

CEN: An appending, 674-byte, fast, direct infector containing the text: 'Hard Disk Failure Lady Seller' and '*.com *.exe ..'. The virus contains a destructive payload; its trigger is based on the system date and includes formatting disks.

```
IVP.674            CD21 7207 E862 00B4 4FEB F5B4 2ACD 2181 F9CC 0773 09B4 098D
```

Jorgito.730

ER: An appending, 730-byte virus. Once a year, on 14 March (beginning in March 1998) the virus displays the usually-encrypted message: 'Jorgitø Was Here Córdoba Argentina'.

```
Jorgito.730        BBD7 F993 CD21 3D83 7874 72BB 4154 438B C305 FE75 CD2F 9380
```

KorWan.1448

CER: A prepending (in COM files), appending (EXE files), 1448-byte (COM) and 1518-byte (EXE) virus which contains the text: '[The Wanderer, June 5th,1994 Korea]'. All infected files have their time-stamps set to 62 seconds.

```
KorWan.1448        909C 3D62 F075 0433 C09D CF80 FC11 7503 E92D 0580 FC12 74F8
```

Lazer.1000

CN: An encrypted, appending, 1000-byte direct infector infecting one file at a time. Amongst other text, the virus contains: '*.com', 'c:\command.com', '*.*', and '-= ≡βLΛZΣR ≡=- (c)'1994'.

```
Lazer.1000         2BCF EB03 90?? ??8A A649 01AC 32C4 EB03 90?? ??AA E2F5 E9AE
```

Nado.584

CR: A stealth, appending, 584-byte virus which contains the text: '[RedViper (c) made by TorNado in Denmark '95]'. The virus displays a red flashing cursor. All infected files have their time-stamps set to 58 seconds.

```
Nado.584           B811 74CD 2181 FB56 5274 53B4 4ABB FFFF CD21 83EB 2690 B44A
```

Nado.602

CR: A stealth, encrypted, 602-byte virus containing the text: '[Undying Lover v1.01][by WarBlaDE/DC '96]'. All infected files have their time-stamps set to 58 seconds. The following is the longest possible template which can detect all infected files.

```
Nado.602           3114 4646 E2FA C33E 8B96 3A02 8DB6 1200 B910 01EB EB
```

Nado.759

CR: A stealth, encrypted, 759-byte virus containing the text: '[CyberBug v. 1.00][made by TorNado DK]Cyberbug.bat'. The virus creates a file 'cyberbug.bat' containing only one line: 'echo > clock$'. Executing such a file destroys current system date and time values and usually crashes the system. All infected files have their time-stamps set to 2 seconds.

```
Nado.759           E800 00CD 01E8 1600 E800 005D 81ED 0E01 E8CE 02E8 4502 E80D
```

Oktubre.1784

CER: A stealth, encrypted, 1784-byte virus containing the text: 'Feliz aniversario Digital Anarchy!!', 'CHKLIST.MS', 'ANTI-VIR.DAT' and 'Virus OKTUBRE Ver. 1.0a By Bugs Bunny [DAN] (c) 26/12/94 Digital Anarchy'. All infected files have their time-stamps set to 40 seconds. On 6 October, the virus overwrites the contents of the first physical hard disk.

```
Oktubre.1784       E800 00B4 FF05 5DF8 72FC 81ED 0A00 1E06 0E0E 1F07 B9A3 068D
```

Pindonga.2072

CER: An encrypted, slightly polymorphic, 2072-byte virus which contains several destructive payloads, including: corrupting CMOS data, overwriting the hard disk, refusing to execute programs from under *Windows*. On 16 and 18 September, the virus may also display the text: 'PINDONGA Virus V1.4. (Hecho en ARGENTINA) Programado por: OTTO (16/9/77) Saludos a: MAQ-MARIANO-SERGIO-ERNESTRO-COSTRA PD: Alguien mate a Bill Gates (El WINDOWS SE CUELGA)'. No simple template for detecting all infected files exists; the following string detects the virus in memory.

```
Pindonga.2072      B403 B102 50CD 1358 FEC6 3A36 0009 7F07 80FD 1074 08EB E932
```

Shoe.1904

ER: A stealth, encrypted, slightly polymorphic, 1904-byte virus armoured with some anti-debugging procedures. It contains a destructive payload. On 1 January, the virus may overwrite first 112 sectors of a hard disk and display the message: 'OOPS .. Sorry For help call now: 555-SHOE or 555-RGNE No rights reserved by M.WEINHOLD'. No simple template to detect all infected files exists; the following string can be used to find the virus in memory.

```
Shoe.1904          3DCE FA75 07B8 AFEC 9DCA 0200 9D9C 2EFF 1E7A 07CA 0200 9C2E
```

Ups.1155

CN: An encrypted, appending, 1155-byte, fast direct infector containing the text: '!\ oH iTs X-MAS /', '*.COM' and '\ThE_UpS-IsT_HiEr/'. From time to time, the virus displays the graphic image of a skull.

```
Ups.1155           8B94 0801 89F3 81C3 4501 B93B 0431 1743 E2FB 5BC3 5E81 EE06
```

V.514

CR: An appending, 514-byte virus containing the text: '*.COM' and '????????COM'. All infected files are marked with the byte 0AAh located at the end of the file.

```
V.514              B900 04F3 A406 1FBA F101 B821 25CD 210E 1F89 EBC3 3D00 4B74
```

V.699

CER: A prepending, 699-byte virus containing the encrypted text: '7.11.V3b'. It corrupts some infected files.

```
V.699              B8FD FFCD 2181 FB11 0775 298C 060C 00C7 060A 00B9 00B4 4CCD
```

V.768

CN: A prepending, 768-byte direct infector, which infects one file at a time. It contains the text: '*.com' and does not infect COMMAND.COM. All infected files have their time-stamps set to 62 seconds. The

virus payload includes a procedure which overwrites the DOS Boot Sector of the current disk.

```
V.768           B9B8 0289 0E90 00A3 9200 A104 002D 8704 7414 2B06 4C00 3D3B
```

V.1097 EN: An appending, 1097-byte direct infector. It contains a payload that includes deleting files with the extension 'zip'. The plain-text ASCII string 'Frvrmfsmvu2/-)v_Hitmru07Vsmsfs$' is visible at the end of infected files.

```
V.1097          E82A FF80 FCFF 740A BAA7 0403 D5B4 41CD 21C3 B4FF C3E8 0042
```

VCC.339 CN: An encrypted, appending, 399-byte, fast direct infector containing the text: 'Marvin the paranoid android'. The payload, which triggers randomly, installs a new Int 21h, which truncates the length of every file loaded for execution to 0 bytes. Infected files have their time-stamps set to 4 seconds.

```
VCC.339         CAE8 1600 EB26 E811 008D 9603 01B9 5301 B440 CD21 E803 00C3
```

VCC.581 CR: An encrypted, appending, 581-byte virus which contains the text 'Mary Reilly'. The virus does not infect files which have their time-stamps set to any of the following seconds values: 36, 38, 44, 46, 52, 54, 60 and 62.

```
VCC.581         4503 8DBE 3B01 BA01 0047 47EB 0590 B44C CD21 B40B CD21 E2F1
```

VCC.613 CR: An encrypted, appending, 613-byte virus containing the text 'The Grim Reaper'. It does not infect files which have their time-stamps set to one of following values: 56, 58, 60 and 62 seconds.

```
VCC.613         E5P7 1581 059E 16F7 1547 47EB 0590 B44C CD21 B40B CD21 E2E5
```

VCC.784 CR: A stealth, encrypted, appending, 784-byte virus containing the text: '*() Mary Mallon = Typhoid Mary)(*'. All infected files have their time-stamps set to 60 or 62 seconds, but the stealth routine ignores the latter.

```
VCC.784         35EC 8FFE 05P7 15F7 1547 47EB 0590 B44C CD21 B40B CD21 E2BA
```

Voyager.1134 CN: An appending, 1134-byte direct infector which does not infect programs 'WI*.*' and 'CO*.*' (e.g. win.com, command.com). The virus contains the text: '*.*', '*.vom' and 'Voyager (.com) is here'.

```
Voyager.1134    80BE 4104 E975 0780 BE44 0421 7413 80BE 4204 5A75 0780 BE41
```

FEATURE 1

Generic Decryption Scanners: The Problems

Carey Nachenberg, Alex Haddox

Anti-virus researchers strive to design their virus scanners to be as general as possible, so that the largest number of viruses can be detected without significant and continuing modifications to the engine itself. This strategy reduces the number of required changes to the anti-virus program, and diminishes the need for regression testing and frequent, expensive upgrade shipments.

This has led to a largely data-based solution to the anti-virus problem. It is quite economical to post a non-executable data file publicly, for clients to retrieve at their leisure. Furthermore, software developers need not worry about software piracy since this data file is useless without the executable portion of the anti-virus program.

Unfortunately, the very nature of computer viruses makes it impossible to design an anti-virus system that can detect current and future viruses without executable updates. New viruses are being developed constantly, and growing numbers use detection-resistant techniques to thwart existing anti-virus algorithms.

Often, anti-virus researchers develop specialized detection routines to deal with these exceptional viruses. However, when enough of these viruses exist, they invalidate the current detection paradigm, and force the development of an entirely new technology. Consequently, the anti-virus software of today is a patchwork of many detection schemes and engines.

Virus writers have already forced many shifts in anti-virus technology. For instance, when anti-virus programs first developed the capability to detect unchanging viruses, the virus authors reacted by developing polymorphic viruses. To detect these polymorphic viruses, anti-virus researchers developed the CPU emulator-based Generic Decryption (GD) scheme.

Now, with the increasingly widespread use of such emulator technology, it is only a matter of time before the virus authors design insidious new viruses to invalidate the CPU emulation technique.

What is Generic Decryption?

Current polymorphic viruses contain at least a small body of machine language instructions and data which is copied verbatim from infection to infection. For the polymorphic virus to avoid detection, this static portion of the virus is

encrypted within infected files. When a program infected with a polymorphic virus is launched by a user, the virus takes control, and launches its decryption routine to decrypt the static portion of the virus. Once this routine finishes decrypting the virus body, it transfers control to the body so the virus can replicate.

The GD scanner relies on this behaviour to detect polymorphic viruses. Each time the GD anti-virus program scans a new executable file, it loads it into a 'virtual computer' (i.e. a simulation of a PC). The program is then allowed to execute in this virtual computer as if it were running on a real machine.

During execution, if the target file is infected with a virus, it can cause no damage to the actual computer, because it executes in a completely contained, virtual environment.

If the GD scanner emulates a program infected by a polymorphic virus, the virus executes its decryption routine. This routine proceeds to decrypt the static portion of the virus within the virtual computer.

As the virus executes, the Generic Decryption anti-virus scanner monitors the progress of its execution. When the virus has decrypted enough of itself, the anti-virus scanner examines these decrypted regions and identifies the strain of the virus exactly.

"the goal of the GD scanner is to emulate as few instructions as possible, while still detecting all infectious virus samples"

The Generic Decryption scanner identifies the virus by searching for specific sequences of bytes which are certain to be present in the static (previously encrypted) portion of the virus. Of course, like other virus scanning technology, the GD scheme requires anti-virus researchers to analyse the virus, extract a virus signature and insert the signature into the scanner database.

In essence, this process is like injecting a mouse with a serum which may or may not contain a virus, and then observing the mouse for adverse effects. If the mouse becomes ill (that is, if the virus manifests itself), researchers can observe the visible symptoms, match them with known symptoms, and identify the virus. If the mouse remains healthy, researchers can select another vial of serum and repeat the process.

Generic Decryption systems provide accurate identification of polymorphic viruses and reduce dramatically the possibility of false identification or misidentification. Such extreme accuracy is possible because the scanner examines the unchanging virus body instead of the ever-changing virus decryption routine.

However, Generic Decryption anti-virus systems are not perfect: there are many ways in which viruses can and do avoid detection by GD-based scanners. The following sections describe several viruses, existing and theoretical, and discuss how they avoid detection by GD scanners.

GD-resistant Viruses

Most polymorphic viruses decrypt and transfer control to their virus body deterministically: a given infection will always decrypt and transfer control to the virus body in exactly the same manner.

As a result, if the viral sample is emulated long enough, the static body will be decrypted and executed, making GD detection possible. However, viruses do not necessarily need to gain control of the computer every time an infected sample is executed.

Consider a virus that uses polymorphic code to fetch a byte from an actively changing area of memory, such as the DOS disk buffers:

- if the value of this byte is between a certain range, then the polymorphic code continues decryption and executes the virus body
- if the value of this byte is outside the required range, the polymorphic code repairs the host program in memory and transfers control to the host program
- every time the virus infects a new file, the location from which the byte is fetched and the required range is randomly changed

This virus might gain control of the machine once in every ten executions of an infected program; however, such a program could still be quite infectious. Unfortunately, the GD scanner is simply unable to detect such a virus reliably.

The GD would emulate the infected sample until it reached the random memory test. If the emulator's virtual memory happened to contain the appropriate value in the proper memory location, the polymorphic code would continue decrypting the virus, and the sample would be detected. If the emulator's virtual memory contained a different value, however, the virus would fail to decrypt itself and the GD scanner would fail to detect the virus.

Given the number of possible memory states (well over $2^{8388608}$ for a simple 1MB PC), it is impossible to guarantee that such a virus infection would always find what it wants in the computer's memory and decrypt itself properly. The Commander_Bomber virus unknowingly employs a similar technique, making reliable GD-based detection impossible.

There exists yet another technique which thwarts GD scanners completely. Generic Decryption requires that the virus gains control and decrypts itself as soon as the host program begins executing. Why? The GD scanner must decide how long to emulate each program before it stops to report that the file is uninfected.

The goal of the GD scanner is to emulate as few instructions as possible, while still detecting all infectious virus samples. To reduce the amount of time spent emulating programs, current GD schemes emulate the suspect program and examine the instructions used by the program in an attempt to determine whether the instructions look like those used by a polymorphic virus.

If the instructions look suspicious, the GD scanner continues emulating the host in an attempt to get the (potential) virus to decrypt itself. If the instructions look like those of a 'normal' program, the GD scanner assumes the program is uninfected and ceases emulation.

Several existing viruses (such as Positron – see *VB*, February 1996, p.8) infect executable files so the virus receives control only after the host program has executed a number of its own instructions.

Thus, when an infected program is launched, the virus may or may not gain control, depending on the nature of the infection. Even if the virus does receive control, it is most likely to do so after one or more instructions of the host program have been executed.

"the virus-writing community is fully aware of GD's inherent weaknesses: it is only a matter of time before viruses which exploit these are constructed"

When scanning such an infected file, a GD scanner would initially emulate the instructions of the host program rather than those of the virus. Consequently, the GD scanner would in many cases recognize these instructions as non-viral and cease emulation almost immediately. The emulator would not emulate the file long enough to reach the virus, hence the virus would fail to decrypt itself, and the file would be reported clean.

The emulator could be set always to emulate many thousands or millions of instructions before reporting that a program is uninfected. However, even with this Draconian modification, there can be no guarantee that the emulator would emulate the host long enough to reach the virus decryption routine.

In fact, there is no guarantee that the emulator would ever execute the instructions of the virus decryption routine, even if the emulation went on indefinitely!

Imagine a program that merely waits for a key-press from the user and then terminates. This program might be infected by a virus such that the virus is given control just before the program terminates. In a typical interactive environment, such a virus would launch every time the infected program was executed by the user.

However, given that the virtual machines used in GD scanners are non-interactive, the program could execute endlessly in the virtual machine, awaiting a key-press from a non-existent user. As the program would never receive a key-press and terminate, the virus would never have a chance to execute and decrypt itself.

GD-pesky Viruses

It is a difficult task to create a fully-compatible CPU emulator. Even a simple flaw that differentiates a CPU emulator from a real machine can be located and targeted by a virus writer.

Even the 80x86 line of computers is not completely backwards compatible. Every processor is slightly different from its predecessors. For example, the pre-fetch queue on the 80x386 chip is sixteen bytes long, but for the 486, the pre-fetch queue was expanded to 32 bytes, in order to increase performance.

Consider an anti-virus product that uses the GD technology with an 80486-compatible CPU emulator. This emulator would be unable to execute properly a virus that employs polymorphic code designed to exploit the sixteen-byte pre-fetch queue of the 80386 processor.

Although it is true that this virus would also fail to execute on real 80486 machines, it might flourish on the large base of 80386 machines. Unless the Generic Decryption implementation applies several different emulators on each file, it will fail to detect this virus. Such a solution is impractical; even if it were implemented correctly, it would increase scanning time significantly.

Conclusions

The Generic Decryption scanning technique has so far proved to be the single most effective method of detecting polymorphic viruses. It allows anti-virus researchers to spend less time analysing specific polymorphic viruses, improves scanner performance, and reduces false positives.

Despite these benefits, GD technology still has significant problems. Many different classes of polymorphic viruses simply cannot be detected reliably. Currently, there are a limited number of polymorphic viruses which employ such anti-detection schemes.

However, the virus-writing community is fully aware of GD's inherent weaknesses: it is only a matter of time before viruses which exploit these are constructed. For this reason, the anti-virus community must remain ever-vigilant and never satisfied with current technology and implementation.

The authors of this article are both anti-virus specialists at *Symantec Corp*. They can be contacted as follows:

Carey Nachenberg: cnachenberg@symantec.com
Alex Haddox: ahaddox@symantec.com

VIRUS ANALYSIS 1

Excel Yourself!

Sarah Gordon

The number of macro viruses appears to increase on a weekly basis, although every new batch seems remarkably similar to the last. However, the most recent macro virus I have encountered requires special treatment: first, it was discovered in the wild; second, it infects *Excel* spreadsheets, not *Word* documents. Do I have your attention yet?

Just as we knew that many vulnerabilities existed in macro languages long before Winword.Concept reared its ugly (but persistent) head, we knew it was only a matter of time before an *Excel* virus appeared in the wild.

Excel spreadsheets have, in much the same way as *Word* documents, the potential to carry code which represents executable instructions in the *Excel* environment. And, also like *Word*, *Excel* does an excellent job of handling these macros; it usually does so flawlessly, without drawing much attention to itself.

Fortunately, Laroux is less than elegant in its design, and *Excel* is liable to notify the user under certain circumstances that the virus' copy routine has failed.

This otherwise unremarkable virus, ExcelMacro.Laroux, differs from its *Word*-based cousins in that it is written using *Visual Basic for Applications* (*VBA*). [*This is a much more powerful macro-ing language than that present in current versions of Word. Ed.*] Clearly, therefore, *Word* users need not concern themselves with this particular virus, just as *Excel* users need not worry about Concept.

The virus carries no deliberately destructive payload, and is only slightly more devious than the first *Word* macro viruses. It uses simple techniques to replicate and hide. Simple... but very effective.

Hide and Seek

As with all new viruses, I began testing cautiously, and opened the infected file I had been sent from a write-protected disk. Immediately, a design flaw in the virus became apparent. A dialog box titled 'Macro Error' popped up on my screen, telling me that a copy had failed, leading me to hope that the virus would prove ineffectual. This hope was misplaced.

[*This box is only displayed when the current drive is write-protected: if the user opens the file by typing 'A:\INFO.XLS', no error occurs. However, if he navigates to drive A, and selects INFO.XLS, the warning is displayed. Ed.*]

Examining the virus proved easy – it could be done either by using Window/Unhide to reveal the hidden Worksheet, then selecting it from the tab display, or by using Tools/Macro to select a macro to edit. Either method could easily be subverted by future such viruses to trigger the virus, just as with Tools/Macro under *Word*.

The first line of the virus code will be familiar to those who have examined *Word* viruses: 'Sub auto_open()'. Although the syntax is slightly different from the *Word* equivalent, the purpose is the same: an Auto_Open macro (*VBA* is not case sensitive) is invoked whenever a spreadsheet is opened.

This particular Auto_Open macro is very simple: it inserts a call to the second virus macro, called check_files, which is executed whenever a new Worksheet is activated. This is the virus' only other macro, and does most of the work.

Check it Out...

When activated, the check_files macro first obtains *Excel*'s start-up path; this was C:\MSOFFICE\EXCEL\XLSTART on my test computer, but will vary depending on your installation. It then looks for the file PERSONAL.XLS in this directory. (Note: users should not assume that the absence of this file indicates their systems are virus-free.)

> *"if the macros 'auto_open' and 'check_files' exist, you are likely to be infected"*

This .XLS file is akin to *Word's* NORMAL.DOT. The *MS Excel/Visual Basic for Windows 95 Programmers Guide* says:

In Microsoft Excel Version 7 you can still record your macros in a workbook that opens each time you start Microsoft Excel ... this workbook is now called 'PERSONAL.XLS' or 'Personal Macro Workbook', depending on the platform (Windows or the Macintosh) ... Microsoft Excel Version 7.0 creates your new Personal Macro Workbook when you record your first macro.

Due to the way Laroux searches for the PERSONAL.XLS file, I suspect it will not replicate on *Macintosh* versions of *Excel*, although no machine was available to test this theory. [*The virus is written in VBA; thus it will also not work on versions of Excel earlier than 5.0. Ed.*]

Laroux next examines the number of Modules (*Excel*-speak for Workbook sheets that contain *VBA* code) in the currently active Workbook (referred to as 'ActiveWorkbook'). There are four possible cases:

- no PERSONAL.XLS file; the ActiveWorkbook contains no Modules. Under normal circumstances, this should not occur: if there is no PERSONAL.XLS, the virus has not infected the host machine, or an error has occurred. .Given that there is no PERSONAL.XLS, the virus

should be running from a macro in the Active Workbook, which would then have to contain at least one Module.

- no PERSONAL.XLS file; the ActiveWorkbook contains Modules
- PERSONAL.XLS file present; the ActiveWorkbook contains no Modules
- PERSONAL.XLS file present; the Active Workbook contains Modules

If the first or last conditions are met, the virus will abort; this serves as an infection check. On an uninfected computer, the second condition is met, and will therefore be considered first.

If the machine does not have a PERSONAL.XLS file, it is not yet infected. It proceeds to unhide the virus Module (titled 'laroux'), and copy it into the PERSONAL.XLS file. It sets some fields in the file properties to empty strings: Title, Subject, Author, Keywords and Comments (why it does this is not clear). This done, the virus cleans up, and infection is complete.

Now, when the user opens an *Excel* spreadsheet, the virus will be activated (the third case). If PERSONAL.XLS exists, and the current ActiveWorkbook contains no Modules, then the virus knows that PERSONAL.XLS is already infected (as the macro is running from there), and it should now infect the active workbook (i.e. the one just opened).

When running from PERSONAL.XLS, the virus watches for new spreadsheets using an 'OnSheetActivate' event. This is more powerful than an Auto_Open, as it is triggered whenever a Worksheet becomes active (i.e. whenever the user clicks on a tab to view a different Sheet within a Workbook). Such routines offer both the macro programmer and the virus writer great flexibility.

Like *Word* viruses, Laroux infects the target by copying its macros there. Unlike *Word* macro viruses, this does not require the alteration of the file's type, but merely the addition of new Modules – in this case, a hidden Worksheet located at the beginning of the workbook.

Once this extra Worksheet has been created, it is tagged as 'hidden', and the user will be completely unaware that anything is amiss. However, there are some cases when this copy may fail, and the virus does not trap these errors. Under such circumstances, the virus will display the error box described above.

Detection and Removal

Determining whether or not your copy of *Excel* is infected is simple. Start the program and select the 'Macro...' option under the 'Tools' menu. If the macros 'auto_open' and 'check_files' exist, you are likely to be infected.

As a second check, select one of these macros and click the 'Edit' button (if the system states that you 'cannot edit a macro on a hidden workbook', unhide the workbook by using the Window/Unhide command). You should see the macros, and a Worksheet entitled 'laroux' should also be visible. Keep in mind that taking all these actions is only valid for this particular virus. Other viruses could render this method useless.

The hex pattern which is given below may be used in conjunction with an anti-virus program to locate the virus, but users should remember to add .XL? to the file extension list. If you suspect that you have this virus, you must check all files, as your users may not always be using the default file extensions.

Simply removing the macros from PERSONAL.XLS and all infected Workbooks clears the virus, although users should remember to remove the 'laroux' Sheet at the beginning of every Workbook. Clearly, both of the detection and removal methods mentioned here are short-term measures: it is to be hoped that anti-virus vendors will implement more efficient fixes shortly.

The Solution

Almost a year ago, when Winword.Concept appeared [*see VB, September 1995, p.8*], I wrote: 'The techniques used by this virus are so simple that any idiot could use them to construct similar viruses. If history is any indicator, we can expect to see more of this type of virus.'

We did see more. We now see that *Excel* viruses are just as trivial; it is safe to assume that there will also be more of them. They will probably be equally unremarkable.

The ease with which these macro viruses can be created and modified means that long-term solutions need to be found soon to the whole threat, rather than to individual instances.

This problem is firmly in the domain of the application developer – they should also keep an eye on the possible misuse of all this extra functionality. It is becoming more and more important as macro virus production increases. Please, designers, pay attention!

ExcelMacro.Laroux

Aliases: None known.

Infection: *MS Excel* spreadsheets. v5.0 or greater.

Self-recognition in *Excel* Spreadsheets:

Searches for a Worksheet named 'laroux'.

Hex Pattern:

```
0021 0060 0027 206A 0020 206A
00AD 0001 005C 0011
```

Trigger: None.

Removal: See text.

VIRUS ANALYSIS 2

Hare Krsna: ISKCON too far!

Ian Whalley

At the end of May, I received a virus sample. In itself, this is not unusual; however, it was not detected by any anti-virus product which the (very computer-savvy) user had, or could obtain. The sample was unpacked, and analysis started.

A swift look at the code revealed that the file was definitely unusual, and very probably infected. Disassembly was complicated by the virus' anti-debugging and -emulation techniques. Once these had been dealt with, the virus was taken apart without too much effort.

Overview

Hare is a multi-partite virus (Master Boot Sector of hard drives, floppy boot sector, COM and EXE files). It is a slow polymorphic (see below for a detailed description), and contains anti-debugging routines. It is encrypted in both files and boot sectors (viruses which encrypt themselves in the boot sector are becoming increasingly common).

The code of this sample, at 7610 bytes, is by no means the longest seen, but certainly makes it the largest non-*Windows* virus in the wild at the moment. This alone provided warning of the effort that would be involved in disassembly.

Functionality

Hare's code is, to say the least, tangled and complex. It uses many interesting techniques, including one which, in my opinion, is extremely dangerous, and could be used in the future by other viruses to greater effect. More of this later.

When the virus receives control from an infected program, it decrypts itself in memory (this involves executing three separate decryptors). Whilst doing this, it issues an Int 21h, AX=FE23h: if AX=000Dh is returned, this part of the virus is present in memory, and installation aborts.

If the virus handler is not present, Hare completes decryption, and installs itself at the end of the MCB chain (using the standard technique of walking the list looking for the Z block). It then passes control to the new resident copy.

Next, the resident component determines whether *Windows* is running, using Int 2Fh, AX=160Ah (Identify *Windows* Version and Type). If enhanced-mode *Windows* is present (including *Windows 95*), it notes the fact for future reference.

The virus then checks a sector on the track one *beyond* the end of the hard disk. This track is sometimes called the Landing Zone, Engineering Cylinder or Test Cylinder, although these terms are somewhat old-fashioned. If this starts with the identifier CCDDh, it is left alone; otherwise, the virus attempts to write a single sector of random data. The routine is flawed, however, and instead of filling a 512-byte buffer in memory with random bytes, it repeatedly places random data into the first word, which is then overwritten with the CCDDh marker anyway!

The data is used by the polymorphic encryption routines to create the header for new instances of the virus; consequently, replicants of Hare created on one PC will have very similar encryption loops. Because of the bug, the effect is not quite what the virus author had intended, but the polymorphic loops will still vary.

This polymorphic technique can foil anti-virus researchers: detectors and removers they create from one infected PC are likely to be incomplete. When an infected program or disk is taken to a clean PC, the random data which Hare writes to the sector will be different, and that PC will create samples of the virus which will look different from those samples which were seen before.

> *"attempts to remove the virus with 'FDISK /MBR' will render the disk unbootable"*

The routine that writes the random data is flawed – when setting the sector number to one (to write to the first sector on the extra track), the virus trashes the top two bits of the track number (which is stored at the top of CL to allow 10-bit track values).

If the disk in question has more than 256 tracks (very likely these days), the sector of random data will be placed somewhere in the middle of the disk, possibly over user data.

Hare then tests to see if its boot sector component is already resident, by issuing Int 13h, AX=5445h. If AX=4554h is returned, it assumes that it is. If it is not resident, it checks the MBR to see if it is already infected, and infects it if not.

MBR Infection

Whilst infecting the MBR, Hare introduces several interesting techniques. It attempts to use port-level access to the hard drive (to avoid BIOS-level boot sector protection).

If it cannot access the hardware directly, it traces Int 13h. It hooks Int 16h (Keyboard) before writing to the disk using Int 13h – it looks as if it is attempting to replace replies to BIOS questions (such as 'A program is about to write to the MBR. Do you wish this write to proceed?') with ones which allow the write to occur. It has not been possible, however, to verify this.

Hare does not leave the Partition Table intact in the infected MBR, so attempts to remove the virus with 'FDISK /MBR' will render the disk unbootable. Later, it must perform complex gyrations (see p.13; 'Loading from an Infected Boot Sector') to account for this.

It is worth noting that Hare correctly uses the *Windows 95* call Int 21h, AX=3513h, CX=084Bh to lock the disk before attempting direct writes. If this is not carried out, *Windows 95* will reject the write. The volume does not appear to be unlocked, but in normal use this should cause no ill effects.

Hare now directly modifies the IVT to revector Int 21h to its own handler – this will enable infection and stealth, of which more later. Next, it checks to see if *Windows 95* is currently running (Int 2Fh, AX=160Ah; returns BH=04h if *Windows 95* is active): if so, it installs its Int 13h hook. Interestingly, it only does this in the presence of *Windows 95*. After performing the actions described under 'Deletion of system file', it returns control to the host program, which is allowed to execute normally.

Deletion of System File

Next comes Hare's most interesting feature. It searches the *MS-DOS* environment data area for an environment variable starting 'WI', which will match either WINDOWS or WINBOOTDIR: these point to the main *Windows* directory on *Windows 95* systems. When this is located, Hare takes the value, appends to it the string '\SYSTEM\IOSUBSYS\HDFLOP.PDR' (thus obtaining a complete path to the file HDFLOP.PDR), and calls the original Int 21h handler to delete it (Int 21h, AH=41h).

Why does it do this? Documentation on the area is limited, but the file HDFLOP.PDR contains the *Windows 95* port-level driver for floppy disk drives. Readers familiar with previous discussions on the impact of viruses on *Windows 95* will be aware that this OS does not normally propagate boot sector infections: it uses direct access to floppy disks, so Int 13h hooks installed by a virus to monitor and infect floppy disks are never called.

Unfortunately, to be able to do port-level access in this way, *Windows 95* requires the file HDFLOP.PDR. If this is not present, the system uses old-style Int 13h access to floppy disks. This is a problem, as now any Int 13h handlers will be triggered, and infection can take place as before.

Worse, *Windows 95* does not warn the user of this scenario: browsing through the contents of the System applet in Control Panel does reveal that the system is not running at peak performance, but normal users do not look here every day.

Thus, after the next reboot, Hare will be able to infect the boot sectors of floppy disks. Better yet, if the virus is removed, this driver file is still missing, and any subsequent boot virus infection will be able to infect floppy disks in the same way.

In Memory: Int 21h

The Int 21h handler intercepts the functions FE23h (Are You There?), 36h (Get Disk Free Space), 4Ch (Exit), 31h (TSR), 00h (Terminate), 4B00h (Load and Exec), 11h/12h (Find First/Next by FCB), 4Eh/4Fh (Find First/Next by Name), 3Dh (Open Existing File) and 3Eh (Close File).

The Get Disk Free Space handler is rather peculiar – when this function is called, the virus checks the address of the calling process's PSP. If it is different from that of the last process which called this function, it performs a genuine Get Disk Free Space call via the original Int 21h handler, saves the number of free clusters, and returns the values unchanged. If it is the same, it still performs the genuine call, but replaces the value for the number of free clusters with the saved one, and returns to the caller.

The reasons for this are not obvious – a couple of possible explanations for this have been suggested. Firstly, one particular anti-virus product, *InVircible*, periodically calls Int 21h, AH=36h to see if the amount of free disk space is dropping. If it detects a drop, it warns that a fast-infecting virus may be in memory. Hare's technique of returning the same value every time the process asks will foil this.

"if the virus is removed ... any subsequent boot virus infection will be able to infect floppy disks in the same way"

The second possible explanation is that Hare is again attempting to fool anti-virus researchers. An oft-used technique for replicating viruses is to place the virus in memory, do a DIR to note the lengths of the goat files and amount of free disk space, run the goat files, and then do another DIR.

Even if the virus has file length stealth, the change in the amount of free disk space will reveal if the virus has infected anything. Hare will not show any change, however, as each DIR command will return the same value for the free space (each call is issued by COMMAND.COM).

Exit, TSR, and Terminate calls are dealt with in the same way: the name of the currently executing program is extracted from the PSP, and that file is opened, infected, and closed.

On Load and Execute, the virus uses a much more complicated handler. After re-deleting the file HDFLOP.PDR, the virus hooks Ints 24h (Critical Error) and 1Bh (Control Break). It then gets, saves, and clears the file's attributes, before going on to examine the filename. It does not infect files whose names match the patterns TB*.*, F-*.*, IV*.*, CH*.*, or COMMAND*.*, nor those containing the letter V.

After infection, the file's time-stamp and attributes are reset (the virus modifies the time-stamp of infected files to set their seconds field to 34), and the handler is complete.

On all Find First/Next calls, the virus can do limited file length stealth – it examines the time-stamp of files encountered and subtracts 7680 bytes from the length of files tagged as infected. This results in infected files appearing to be larger than they were before, albeit not as much so as they actually are.

The Close Calls functions invoke a handler which will infect the file if it is deemed necessary – it first extracts the filename by manipulating the SFT (System File Table), performs the name checks described above, and then, where applicable, infects.

On Open Existing File requests, if Hare determines that the file is infected, it is disinfected (on disk) before the call is allowed to proceed. This will temporarily remove the virus from the file in question, which will be reinfected when the file is closed.

In Memory: Int 13h

The Int 13h handler performs stealthing of infected boot sectors, and also infects the boot sectors of floppy disks as they are used.

This latter is accomplished by first ensuring that the floppy in question is not already infected – the virus reads the boot sector (it retries three times; just what the manuals say should be done), and subtracts the word at offset 100h from that at offset 102h. If the result is CCFFh, the boot sector is deemed infected.

If the floppy is not already infected, Hare formats an extra track at the end of the floppy disk, encrypts the virus code, and writes the body to the extra track, and the loader code to the boot sector.

Loading from an Infected Boot Sector

When a computer is booted from an infected hard drive, the virus shuns the standard approach of immediately installing an Int 13h handler – this would make life much easier for it, as its own stealth features would allow DOS to see a valid partition table.

Instead, Hare copies the partition table back into the MBR whilst loading; thus, when the OS loader comes to look, the partition table is where it is supposed to be. It then knocks 9KB off base memory by the standard technique of modifying the word at 0000:0413h, intercepts Int 1Ch (System Timer Tick), and passes execution to the code of the original Master Boot Record.

Using a technique already seen in several other viruses, Hare monitors Int 1Ch to watch the operating system load. When it determines that it is safe to do so, it intercepts Ints 13h, 21h, and 28h (DOS Idle Interrupt). The first time the system issues an Int 28h (which will happen as soon as a program waits for input), Hare re-corrupts the partition table (which was fixed to allow the OS to load). It is now in a position to infect files and disks as they are accessed.

Trigger

On 22 August and 22 September, the virus' trigger routine is activated. First, it displays the message:

```
"HDEuthanasia" by Demon Emperor: Hare Krsna,
hare, hare...
```

Next, it attempts to wipe all data from all hard drives on the system with garbage.

Conclusion

Despite the many new and interesting techniques displayed by Hare.7610, the virus has several bugs. It is generally unstable, and replications will sometimes not execute properly (this includes MBR infections), and will hang the machine. The destructive trigger also sometimes fails. The fact remains, however, that Hare is in the wild across the world, and appears to be spreading. So far, it has been found in the wild in Canada, Russia, the Netherlands, Switzerland, the UK, and the USA: it appears that such a wide distribution was achieved via the Internet.

[*Note: Two variants of Hare.7610 have been discovered, Hare.7750 and Hare.7786. As well as bug fixes, they will occasionally (one in sixteen times the system is booted from an infected disk) change the random data sector. This means that the polymorphic algorithm will change periodically on any given computer. Hare.7750 was distributed via posts on the Usenet groups alt.cracks, alt.crackers, alt.sex, and alt.comp.shareware.*]

Hare.7610	
Aliases:	Krsna, HDEuthanasia.
Type:	Slow, polymorphic, multi-partite virus.
Self-recognition in Files:	Seconds field of time stamp set to 34.
Self-recognition in Boot Sectors:	Word at offset 102h in BS minus word at offset 100h equals CCFFh.
Self-recognition in Memory:	Int 13h, AX=5445h, expects return of AX=4554h. Int 21h, AX=FE23h, expects return of AX=000Dh.
Hex Pattern:	None possible.
Intercepts:	Int 13h, 16h, 1Bh, 1Ch, 21h, 24h, 28h.
Trigger:	On 22 August/September, prints message and attempts to trash disks.
Removal:	Identify and replace infected files. Format infected diskettes. Replace hard disk MBR with known clean copy (FDISK /MBR must not be used).

FEATURE 2

Viruses on the Internet

Sarah Gordon

Author's note: This article explores attitudes to virus distribution facilitated by the Internet. Our increased reliance on the Internet for communication, and the retrieval of information from untrusted systems, can be expected to bring more cases of point-and-click giving users new viruses of many types, including those which take advantage of existing security holes in insecure applications.

The World Wide Web is a wonderful place. In June 1996, I decided to explore it to research this article; specifically to gauge the success of the 1995 'let's get rid of Internet virus sites!' campaign which had been sponsored by the *NCSA* and some anti-virus product developers.

My first search brought me fifty thousand matches. After regaining my composure, I realised many of these must be related to other types of virus. Fortunately, a narrowed search proved I was right. Surely we are winning the battle to encourage responsible behaviour on the Internet!

Or are we? With my refined search, I found 2000 matches to computer and virus (or virii, as virus distributors like to call them). The first site I came across was one that offered the classic 'computer virus joke' file:

```
Arnold Schwarzenegger Virus. Terminates, stays
resident. It'll be back.
Freudian Virus. Computer becomes obsessed with
marrying its own motherboard.
Star Trek Virus. Invades your system in places
where no virus has gone before.
```

What was to come was not so amusing. As I pointed and clicked, I found other 'virii' sites. Some pages were not fully operational, but many more were. Some were old pages I had run across months ago which had been taken down during the brief flurry of 'stop the virus sites'.

At that time, I predicted that the sites would come back, or reappear under other names. I hate to say it, but... *I told you so*. The sites have returned, and the methods we have tried to use to stop them have not worked.

Anatomy Lessons

What exactly can be found by following the downward spiral of the World Wide Web? More than some people would have you believe, to be sure.

I began with a site reference on university coursework. This was of particular interest to me, as I had just returned from the IFIP Conference in Samos where I heard a Swedish professor explain that making viruses was part of his curriculum. When I mentioned that two of the virus writers with whom I had spoken were students at his university, he told me he had heard about them, but he did not seem to think it noteworthy.

The following, a description of coursework from an American university, illustrates the casual attitude toward viruses which seems to prevail at many universities.

> *Computer Virus analysis*
>
> *Take a computer virus and analyse it thoroughly. You will have to isolate the virus code and disassemble it ... Once you have it disassembled, you now have a program listing which IS the virus. Go through it, one assembly language statement at a time, and figure out what it does and how it works. It is best to do this on a fairly simple virus ... I have a copy of the Natas virus if you want to try that one.*

This was the most responsible entry. While some would say using viruses as part of a learning exercise is 'good experience', others say it is 'poor science'. Deciding whether or not Natas is a 'fairly simple virus' remains a task for the reader. From this site, it was all downhill.

Under the banner 'Free Speech On-line Blue Ribbon Campaign', I was welcomed to 'The Virus Page: VIRUS PROGRAMMING and VIRII'. I was invited to join the Blue Ribbon Anti-Censorship Campaign and given access to all sorts of virus tutorials. There was information on disinfecting infected files, TSR, COM infections, non-overwriting COM infections, infection on closing, EXE infections, directory stealth, memory stealth, and a memorable tutorial, 'The Dangers of Thunderbyte'.

Polymorphic viruses were part of the plan as well, with 'Implementation, Detection, and Prevention'. Other instructions included infection of *Windows* executables, calling *Windows* API in assembly language from VLAD, heuristics, ANTI-AV Tricks (Tunnelling), Inbar Raz's Guide to Anti-Debugging Techniques and (from our own side), 'Anticipated trends in Virus Writing – Some ideas from the AV folks'.

There were also assembly language links, programming tools including A&6 assembler v4.02, A&6 debugger, a 32-bit *Windows* disassembler, *ViruScan for Windows 3.x*, *TBAV for Windows 3.x*, and, to my utter horror, *F-PROT*.

Does anyone actually get anti-virus software from sites which offer the latest and greatest virus source and executables right alongside anti-virus software? You would hope not, but I learned that some people do!

Some company employees of major firms told me that they 'trust' the virus sites because there is so much 'information' there. These are the people who are responsible, in some cases, for securing your systems. There were links to other

pages, too numerous to mention, most of them virus-related. There was even a link to Alan Solomon's hacking and virus laws page.

A trip to one of the links showed the same viewpoint, or possibly pseudo-viewpoint, one I saw repeated many times:

> Disclaimer: These files are for research and educational purposes only. I take no responsibility for any misuse of these programs which can result in ARREST OR DAMAGE TO YOUR COMPUTER. Please keep in mind that viruses are harmful and may destroy your computer: if you destroy other people's computers, you will be held responsible. Download at your own risk!

That site had files. The files were viruses, nicely catalogued. It also had generators, constructors and source code files. The warnings are nice. But who's kidding whom? Virus distribution in this manner is nothing less than irresponsible.

When I asked some of the people involved, the responses were generally that if the person who downloaded the viruses was incompetent to manage them, it would be that person's problem; that it is always the user's own choice to download. Virus sites are well and truly on the Internet, and they are here to stay.

"there are real problems in becoming the censor of user communications, both from a legal and an ethical standpoint"

A Problem with the American Legal System...

...is the outcry of some anti-virus researchers. Indeed, this is a possibility worth considering. People may take this position because some American-based public Internet Service Providers (ISPs) and on-line services hide behind the whimper 'it's not illegal'. Does this demonstrate a terrible ethnocentricity on the part of these providers? After all, the Internet is global.

An examination of one of these same providers' publicly available FTP logs shows computer viruses being siphoned to the UK just last week. Japan is another popular location on the receiving end of viruses from American ISP clients.

However, is action on the part of the service provider part of the solution? Is 'it is not illegal' adhering to the outdated paradigm 'If it's not illegal it must be OK'? Some would argue that it is, and that ISPs and on-line services should take more responsibility for the actions of their users and for the welfare of the computing public. Others recognize that there is, in fact, no viable solution.

There are real problems in becoming the censor of user communications, both from a legal and an ethical standpoint. These problems place ISPs, on-line providers and bulletin board operators in situations which may be impossible to resolve.

In 1994, representatives of several unnamed commercial ISPs and on-line services were questioned by various people regarding their policies on allowing viruses to be distributed or made available from their servers[1]. Reactions varied from 'it's legal' and 'we cannot become censors of our users', to 'we will not knowingly allow such things to be made available on our site'. It is interesting to note, however, that all the sites queried still have viruses and other 'questionable' material available from time to time.

Of course, service providers' views are based not only on the laws, but on the feelings of their customers and potential customers. 'Is it OK to make viruses available for public consumption, via the Internet?' – I have asked this question countless times, in public forums, on BBSs, at Conferences. Opinions seem to fall into two categories:

- it's nobody's business what anyone else does as long as it doesn't hurt anyone directly
- you can't do that because I don't like it

Defining 'directly' seems to vary from culture to culture; that discussion is best left for another publication.

I thought it might be interesting to query individuals in the IT field and ask the same question. The responses reflect what I have heard from the computing community in general. Only two responses stated that virus distribution should be illegal. The first said:

> *Maybe virus distribution should be illegal, but policing it will always be a problem. The Internet offers a new perspective on the 'Global Village' concept. These are issues yet to be resolved – who knows if they ever will be? A person who makes viruses available should share the responsibility, but the key word is 'should'. That opens a new arena of conflict: we must learn to be wary and learn how to avoid these problems. The ideal would be nice; people providing only helpful, useful items on the Internet. There should probably be some sort of punishment for malicious intent, but I hesitate to invite excessive government regulation to the Internet.*

A similar response:

> *I don't believe in censorship in many cases. I do believe in restricting the public market. If a person wants to write a virus, he should have the freedom to do so. If he wants to send it to his friends, still his business. If he would like to place it on his own FTP site and distribute it, as long as it is clearly marked as virus, then he should be allowed. Any distribution of the virus into the public should be illegal.*
>
> *It is the responsibility of the individual if he is on the Internet to watch out for harmful code. It should be assumed that files being downloaded may be infected.*

Then, there were those who took a more casual attitude:

> *Since I've never had a virus, and don't work on systems that most viruses infect, I'm just not that familiar with, or interested in, viruses. I find that most*

people who are very interested in viruses are those who got one and were determined to 'out' their intruder, to figure out everything they could about the creator or the processes involved.

I have a Macintosh at home. I am not very concerned about getting a virus at home, though I use Internet services daily (I don't use BBSs at all though). I run Disinfectant occasionally, but more out of a sense of duty, than fear. I don't have Word, or any other (known) macro infecting program. I think as these things go, based on my user habits and stuff, I have a low propensity for actually getting a virus. But I may be wrong.

These views seem typical of most Americans I have queried, but, despite the claim you will often hear that the USA distributes all the viruses (it used to be Bulgaria – I suspect neither deserved the amount of 'credit' bestowed upon it), I found virus distribution on the Internet to be culturally diverse. The US was there, but along with Canada, Austria, Portugal, Germany, Sweden, Norway and the UK. Viruses were available via FTP, WWW, or in casual trading centres such as IRC: they seem to have become the POGS of the Information Age.

New acquisitions are made with relative anonymity and virtually no interference. The logs of a real server, recorded 1–18 June 1996, showed various viruses, including Monkey and variants of Stealth, being retrieved by willing users. It is possible, of course, to identify users who obtain viruses via anonymous FTP or WWW should one desire to do so.

IRC BOTS dispensing viruses seem to have, at least for now, disappeared. I was pleased to hear this, but then reminded by a cynical friend that there was no need for VirusBOTS. After all, why spend the time getting limited information from a BOT when you can get all the viruses, source, and tools you want directly from the World Wide Web?

We still have the question 'How can we prevent this sort of irresponsible behaviour?' The problem seems to be that we don't really know whom we should be asking to stop it. Although, for the most part, virus download areas eventually fall into disrepair and disappear, there is a continual influx of 'young blood', keeping the number of sites in some sort of steady state.

The ISPs, companies, or universities which host these sites will not, for the most part, stop allowing such activities. For every site which acts responsibly, and does prevent such behaviour, there is a person determined to exercise his rights, oblivious to the concept of duty and responsibility...

As the college has taken this page away from me, I am searching for a new home for this information. Please, if you have any suggestions, email and tell me, I'd like to make the page available as soon as humanly possible. I'm sorry about this, but don't let it discourage your learning, because I won't let it discourage mine.

-The Demon X(a/n)^th

Supply and Demand

Who are the people commonly said to share in the Vx Internet pie? The four groups in contention for this dubious honour appear to be the virus writers and distributors themselves; the average user; the employee (who may be in charge of tech support or product evaluation); and finally, the anti-virus product developer.

The group with the most potential interest in VxWWW sites are the virus writers and distributors themselves[2]. Much of the information stored on such sites is of reasonably high quality, and can provide interesting pointers (in the form of source code or text files) to new techniques. For those who trade viruses, the attraction of such sites is obvious.

How much impact these sites have among virus writers is questionable; however, in the same way that a frisson of fear went through the industry when the VxBBSs began to appear (though the boards had little discernible effect), it is entirely possible that the impact of viruses on the WWW will not lead to vast numbers of new viruses or variants. Only time will tell.

"making viruses available via the Internet may be the 'right' of some people in some countries, but it is not responsible behaviour"

The second group, which encompasses the average user, is in the unenviable position of having the intrigue of viruses thrown at him by the media, the scare put into him by some companies, and the WWW at his disposal to get 'information' which he may think will help him protect himself.

What he does not realise is that this point-and-click could cost him his data: infected documents and Trojanised information abound on the Internet. The biggest risk which is posed to the 'average' user by these boards is that of accidental infection.

The third group with an interest in VxWWW sites comprises those interested in obtaining viruses for product testing. Although some anti-virus companies have gone so far as to recommend this, such actions are demonstrably wrong. After all, without investing a significant amount of time and expertise, it is next to impossible to verify a virus collection obtained from a third party, or to remove all Trojans, joke programs, first generation samples, simulated viruses and corrupted files.

Tests carried out on a virus collection which is not clean (i.e. does not contain real viruses) are meaningless at best, and can be completely misleading[3]. Thus, these sites are of little use as a source of scanner fodder; the problems outstrip any possible benefits.

The final group, the anti-virus product developers, are presented with a unique situation. Ever since the beginning of 'public' virus distribution, the mainstream anti-virus industry has scorned those who trawl the boards for the latest viruses. This was done initially because many VxBBSs required a user to upload new viruses to gain connect time, and also to prevent the legitimization of particular boards. However, the issues are no longer as clear.

At the recent *NCSA IVPC Conference* in Washington, one anti-virus company spokesperson publicly admitted obtaining viruses from Vx sites. I am totally against irresponsible virus distribution and joined with the majority of vendor representatives who chastised the errant company.

However, we do need to keep up with virus authors: accessing what they make available to the general public, to our customers. Knowing that people are in fact accessing and experimenting with these viruses may force a change of heart among the anti-virus community.

I believe much of the anti-virus community's reaction to the admission by the unnamed company was overreaction, based on our instinctive distaste for Vx sites in general. It is one thing to say you do not condone them while sneaking around giving or receiving viruses; unfortunately, some vendors are said to have been involved in this.

It is another matter altogether to admit that, due to the proliferation of these places, we must keep up with current trends. The only way to do that, some say, is to see what is there; to access and examine the viruses.

Unlike the VxBBSs of old, the viruses are there, free for all, only a point-and-click away... what are we supposed to do? Most anti-virus researchers do not obtain viruses from these places, claiming the mixed messages this would send outweigh the benefit of ethical behaviour related to viruses on the Internet. However, the issue is much less clear-cut than you might believe.

Clearly, the Internet is a fabulous place to obtain viruses, no matter who you are or what your intentions. Granted, you shouldn't use them to test anti-virus software. Such tests have been shown many times over to be flawed, and in some cases dangerous to the health of your company. You should not spread them to the unwilling and unknowing – even most virus writers acknowledge this. There is nothing a user can 'learn' from looking at viruses which cannot be learned from non-replicating programs.

Unless you are a product vendor or virus writer, the benefit to you from such sites is practically nil – and even if you are a vendor, the benefit is limited. The risks these sites provide to computer users in general, however, remain high. Owners and maintainers of such sites have no control over how the materials they make available are used. While this is the case with most FTPd or WWW materials, it is particularly undesirable in the case of viruses, as they are uncontrollable once released.

This leaves us with the question, again: 'What is the purpose of allowing such irresponsible behaviour?'. Maybe you believe it is an exercise in free speech, or that it is a 'right'. Making viruses available via the Internet may be the 'right' of some people in some countries, but it is not responsible behaviour. It is also, unfortunately, not showing any signs of slowing down.

Closing Thoughts

Finding a suitable conclusion to this article has been difficult, because I don't think that we are even close to finding answers. We don't know whom we should ask such simple questions as 'Why do we allow this kind of irresponsible behaviour on the Internet?'.

While it is a cliché to say that the Internet causes us to re-evaluate what we mean by censorship and freedom of speech, there is little doubt that the rapid development of the WWW has outstripped our ability as a society to control its contents.

Yes, there are viruses on the Internet, accessible via the World Wide Web, FTP, IRC, email, Usenet and other ways not discussed in this article – but we must keep our perspective. There are also infinitely more threatening problems, like child pornography, which I was unfortunate enough to encounter during my research for this article. The issues to which the Internet gives birth are much bigger than simply computer security and viruses. They envelop our communications with the fabric of cultural diversity, and force us to change the way we, in our own hometowns, think, live and do business.

There is no easy way to make us all think in the same way and magically solve the problem of irresponsible action on the Internet, be it child pornography, church-burning sound files, or computer viruses. We who work to fight computer viruses can only try to educate the public to protect itself from those who put the responsibility on the 'other guy'.

It is possible that, someday, those who view it as incumbent upon the 'other guy' to be technically competent, responsible, and ethical will realise that individual responsibility begins with not distributing or writing computer viruses in the first place.

Footnotes:

[1] *Virus-L Digest*, Fridrik Skulason. August 1994.

[2] 'Technologically Enabled Crime: Shifting Paradigms for the Year 2000.' Sarah Gordon. *Computers and Security*. November 1995.

[3] 'Analysis and Maintenance of a Clean Virus Library.' Vesselin Bontchev. *Virus Bulletin Conference Proceedings*. September 1993.

The views expressed in this article are those of the author, Sarah Gordon, a researcher at *Command Software*. Readers wishing more information on the subject may contact her via email at: sgordon@low-level.format.com.

PRODUCT REVIEW 1

CPAV for NetWare

Martyn Perry

Having recently evaluated *Norton AntiVirus*, this month we look at its stable-mate, *Central Point Anti-virus for NetWare* (*CPAVNET*). This supports both version 3.x and version 4.x of *NetWare*.

The product is licensed on a per server basis, and the software for workstation protection requires separate licensing. Although perhaps more applicable to workstation licences than to servers, the user may have the software on a single home computer, provided that the software receives at least 80% of its use on the primary computer.

CPAVNET comes with manuals for DOS, *Macintosh*, and *NetWare*. In addition, there is a manual for the alert management software, Central Alert.

Installation

Installation is performed in three stages. First, the workstation, which is used to install the network software, is checked and a version of the DOS product is copied to it. Next, the NLM is installed from DOS. Finally, the control and configuration software is installed either to the workstation (for local use) or onto the network for access (from any workstation). Both *Windows* and DOS versions of the control program can be installed into the same directory, \CPSNET. A nice feature is the display listing the files to be installed, highlighting the file currently being copied.

Multiple servers can be grouped together into one or more 'security domain'. The file servers to be grouped together into a domain can be selected individually, provided that sufficient licensed copies of the software are available. The domain name can be freely chosen.

When the NLM is installed, its files are copied from the first disk to the server. These include the directories SYS:SYSTEM, SYS:SYSTEM\CPAVNET, SYS:SYSTEM\CPS, and SYS:SYSTEM\CPS\CALERT.

The installation process next offers to add lines to AUTOEXEC.NCF to load the NLM at server boot time. There is a prompt to LOAD CPMASTER on the console, to allow the administration or configuration program to be run. The installation finally offers the chance to install the Configuration Program for DOS, *Windows*, or both.

Loading the NLM

If the automatic load option is not chosen, the *CPAVNET* NLM program is loaded from the server console prompt using the command 'LOAD CPAVNET.NLM'. This loads the main NLM plus a number of subsidiary NLMs. CPMASTER.NLM must also be loaded on at least one server in the domain, to configure the various options for the scanner and activate Central Alert.

The CPAVNET.NLM can be driven from the server console using the function keys to start or stop an immediate scan and to enable or disable the NLM. Additional function keys allow for keyboard locking and for the application of password protection.

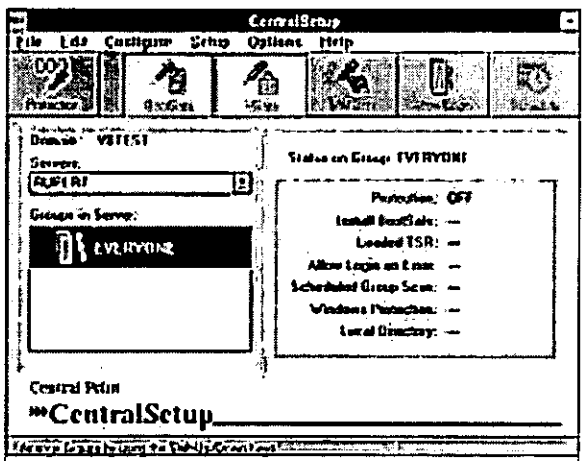
Administration

The scanner administration can be managed from the DOS or the *Windows* configuration program running on a workstation. The CPMASTER NLM must be running on each server to be configured before the configuration program is run.

The software allows servers to be added to a security domain, providing that the administrator has the necessary supervisor rights to those servers. The main administration screen provides access to view the various servers and their protection status. *CPAVNET* has the usual three modes of scanner operation: immediate, real-time and scheduled.

An immediate scan checks the server on demand, using the current immediate settings. Scanning on the server can be started either from the option on the workstation, or by using F6 on the server console.

The on-access, or real-time, scan allows scanning to be performed when a file is copied to or from the server, or when a file on the server is otherwise accessed. It is not possible to disable this option completely; scans of incoming or outgoing files, or both, must remain selected.



The setup program allows installation to be customized in various ways.

A scheduled scan provides scanning on a timed basis. An additional option is to have periodic scanning, which occurs at regular intervals; e.g. every hour between a defined start and stop time. It is possible to start another NLM after a scheduled scan is completed – for example, a backup NLM could be executed here.

Configuration Options

For each mode of operation, various selections can be made. These include the file extensions to be included in the scan: the defaults are EXE, COM, DLL, OV?, SYS, BIN, 386, FON, ICO, and CMD. Extensions may be added or removed as necessary.

As well as file selection, the product provides the ability to exclude files from the scan. This exclusion from on-access scanning is the only way in which infected files can be handled manually.

A separate menu option allows the selection of actions to be taken upon detection of a virus. There are three choices here; to delete an infected file, to move an infected file to a user-defined quarantine directory (the default directory is SYS:SYSTEM\CPAVNET\INFECTED), or to do nothing with the file.

An extra option is included, which *Central Point* defines as analysing for unknown viruses. This examines a file for 'suspicious behaviour'.

Alert Management

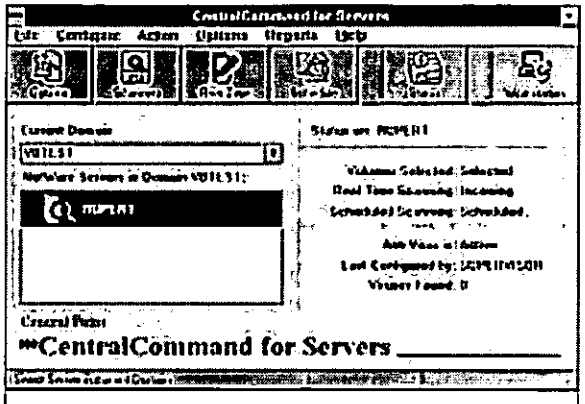
Apart from the action items which occur on virus detection, there is a separate alert program, Central Alert. This will allow modification of the current security domains as well as the sending of alerts to various alert facilities. These are:

- alphanumeric or numeric pager
- *NetWare* broadcast to the workstation
- flash Central Alert icon [*! Ed.*]
- log alerts to a file
- send MHS mail
- send SNMP traps to *NetWare* Management System workstations

Reports, Activity Logs and Updates

CPAVNET keeps a record of events in an Activity log. The events to be logged can be chosen and include:

- detection of known and unknown viruses
- scanner start and end times
- action taken
- enabling and disabling of *CPAVNET*
- loading and unloading of *CPAVNET*
- virus signature changes
- miscellaneous errors and warnings



The administrator can choose which type of scan to have running.

With this amount of data some sort of control is needed, which is supplied in two ways; first, by limiting the size of the log file, second, by filtering the events being displayed. There is a problem here – the whole log file needs to be loaded before it can be filtered. Other information available includes a virus list, and domain status (down to individual servers and their status).

Updates are performed by selecting the appropriate compressed file (for DOS, NLM, etc), and copying it to a temporary directory on the workstation. From here it can be self-extracted and resultant files copied to the correct directories.

Detection Rates

The scanner was checked using In the Wild, Standard and Polymorphic test-sets. Undetected viruses were identified by using the 'delete files' option and listing the files left in the virus directories. The tests were conducted using the default scanner file extensions supplied.

The results were generally disappointing. The tests were initially performed using the virus signatures shipped with the main product (March 96), then using the latest available (June 96). The Standard set scored 37.2% on initial scan: the updated version achieved 60.4%. The In the Wild set managed 65.7% on both passes, which implies that no detection improvement was made in the three months between the signature updates. The Polymorphic result improved slightly, from 41.4% to 43.5%, by virtue of the scanner finding additional instances of the One_Half virus.

A further scan was performed with the Virus Analyzer option selected. This made no difference to the results, however.

Real-time Scanning Overhead

To determine the impact of the scanner on the server when it was running, the usual tests were executed; copying 63 files of 4,641,722 bytes (EXE files from SYS:PUBLIC) from one server directory to another using NCOPY. The directories used for the source and target were excluded from the virus scan to avoid the risk of a file being scanned while waiting to be copied.

Because of the different processes which occur within the server, the time tests were run ten times for each setting and an average was taken. The chosen tests were executed in two groups, for two conditions.

The first group was run with on-access scanning selected first for both incoming and outgoing files, then for incoming files only. The tests were first run without the Analyzer, to establish the effect of the scanner by itself on the server performance. The four tests were:

A. NLM not loaded: this established baseline time for copying the files on the server

B. NLM unloaded: this test was run after the other tests to check how well the server returns to its former state

C. NLM loaded, using default setting of on-access scanning for incoming and outgoing files – immediate scanner not running. This tests the impact of on-access (real-time) protection.

D. NLM loaded, on-access scanning for incoming and outgoing files – immediate scan running. This shows the full impact of running the scanner on server files.

The tests were repeated with the Analyzer selected to judge its impact on performance. A separate set of tests was run with on-access scanning set for incoming files only.

At first glance, the results look a little strange. The difference in time between incoming/outgoing scan and incoming only scan were within the process variability of the server and, for practical purposes, can be viewed as the same.

The results with the Analyzer on appear to be better than those with the Analyzer off. Again, this could be attributed to server process variability; alternatively, it may indicate that separate buffering is used to process the file under analysis, leading to a slightly improved performance.

The performance overhead of checking files using the Analyzer does not appear to be significant. However, in view of the lack of additional success on the test machine, it is debatable whether or not this feature is useful. *CPAVNET* performs a clean unload of all the files which were originally installed, so there is effectively no overhead.

Conclusion

The product is easy to install and performing upgrades is straightforward. The documentation provided is clear and comprehensive.

CPAVNET's scanner detection rate is, and has been for some time, at a level unacceptable for a mature product. It is sad to see a product, which is 'feature rich' in other aspects, fail so badly in this crucial area. This product cannot be recommended as a first-time purchase due to this basic weakness. Existing users should consider biting the bullet, and take the opportunity to move to a product which is better supported; otherwise, they leave themselves seriously exposed to new virus threats.

Central Point AntiVirus for NetWare

Detection Results

Test-set[1]	Viruses Detected	Score
In the Wild	197/300	65.7%
Standard	247/409	60.4%
Polymorphic	4141/10000	41.4%

Overhead of On-access Scanning:

Tests detail the time taken to copy 63 EXE files totalling 4.6MB. Each test is carried out ten times, and an average taken.

	Time	Overhead
NLM not loaded	10.7	n/a
Incoming/Outgoing; Analyzer Off		
NLM loaded, no manual scan	16.2	51.0%
NLM loaded, manual scan	44.8	319.0%
Incoming/Outgoing; Analyzer On		
NLM loaded, no manual scan	16.6	54.0%
NLM loaded, manual scan	43.8	309.0%
Incoming only; Analyzer Off		
NLM loaded, no manual scan	16.4	53.0%
NLM loaded, manual scan	46.0	329.0%
Incoming only; Analyzer On		
NLM loaded, no manual scan	16.3	52.0%
NLM loaded, manual scan	44.5	315.0%

Technical Details

Product: *Central Point AntiVirus for NetWare.*

Developer/Vendor: *Symantec Corporation*, 10201 Torre Ave, Cupertino, CA 95014, USA. Tel +1 408 252 3570, fax +1 408 253 4992.

Distributor UK: *Symantec UK Ltd*, Sygnus Court, Market Street, Maidenhead, Berks, SL6 8AD. Tel + 44 1628 592222, fax + 44 1628 592393.

Price: The per-server price of this product in the UK is an estimated £600-£645. For site licences, apply directly to the company's corporate accounts division in the UK: Tel +44 1628 592222.

Hardware Used: Server – *Compaq Prolinea 590* with 16MB of RAM, 2 GB of hard disk, running under *NetWare 3.12*. Workstation – *Compaq 386/20e* with 4MB of RAM, 207 MB hard disk, running under *MS-DOS 6.22*, *Windows 3.1*.

[1]**Test-sets:** For a complete listing of all the viruses used in this review, see *Virus Bulletin*, July 1996, p.22. For a complete explanation of each virus, and the nomenclature used, please refer to the list of PC viruses published regularly in *VB*.

PRODUCT REVIEW 2

Survival of the Fittest?

Dr Keith Jackson

The *AllMicro Anti-Virus Survival Kit* (*AVSK*) claims to have 'four levels of defense to help keep your PC virus-free and your data safe'. Versions of *AVSK* suitable for DOS, *Windows 3.1* and *Windows 95* are included, though this review covers no *Windows 95* features.

Levels and Features

The *AVSK* manuals make great play of the fact that various 'Levels of Defense' are included. Level 1 incorporates a scanner, memory-resident software, checksumming features, and facilities for disinfecting infected files. Levels 2 and 3 are software updates, and the response of the developers to new viruses reported to them. Level 4 refers to data recovery facilities which can replace a damaged boot record and/or primary partition.

Although I cannot think of many anti-virus software developers who do not offer software upgrades and responses to new viruses, and the majority of anti-virus products incorporate 'data recovery' features, this is not meant to decry the features available within the software itself.

Included with the version provided for review are DOS, *Windows 3.x* and *Windows 95* versions of a full-featured menu-driven interface, a command-line-driven scanner, two distinct types of memory-resident software, a utility which reports on system facilities, and even a communications package which can be used to obtain software upgrades and/or virus signatures. There are too many components to discuss individually, so why dress things up by wittering on about 'Levels of Defense'?

Documentation

The printed documentation comprises a 125-page A5 *Users Guide*, and a 40-page A5 *RESCUE Users Manual*. A statement on the first page of the *Users Guide* reads that it 'avoids technical details'. This is true. Very true.

Sad to say, I found myself unimpressed by the *Users Guide*. It has a tendency to descend into trite explanations. For instance, is the explanation 'Mouse Active – activates or deactivates the mouse' really helping anybody?

The explanations of what to do if a virus is detected are sketchy, to say the least. This is somewhat offset by the on-line documentation, which provides information about individual viruses: short explanations of what the virus can do, presented as a series of boxes, reminiscent of *NAV*. It is, however, not enough. However, hardened users need more detail, and new users need more explanation.

On the plus side, the switches used by the command-line-driven version of *AVSK* are all listed in the *Users Guide*, along with an accompanying explanation. Similarly, all available options for the memory-resident programs are also thoroughly explained.

I have more time for the *RESCUE Users Manual*: although short, it provides a decent explanation of the data recovery facilities provided. Once again there is no index, but this makes less difference in a slim volume.

Installation

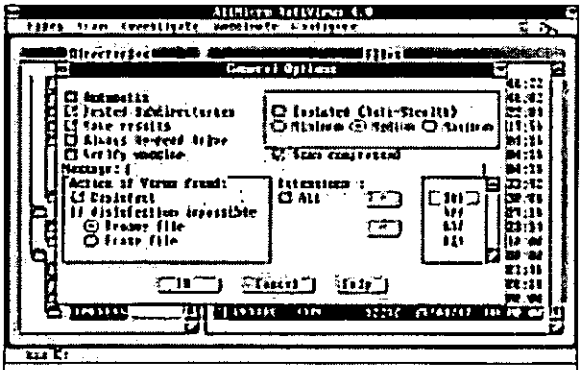
AVSK arrived on four 1.44 MB, 3.5-inch floppy disks, none of which were write-protected. Installation of the DOS version proved straightforward. After an initial warning message about viruses has been displayed, the installation offers to place the *AVSK* files in the default subdirectory, C:\AMAV – this can be altered to any desired location. The user's name and company must be entered to personalize installation; memory is then scanned: if no viruses are found, the *AVSK* files are copied across to the hard disk.

At this point, users are asked: 'Do you wish SENTINEL to be run from your AUTOEXEC?'. On-screen explanation would be more helpful – SENTINEL is a memory-resident scanner. The installation program next advises that the first action should be to create a SAFEDISK (for rescue purposes). Installation is then complete.

Installing the *Windows* program proved to be even simpler. The SETUP program offered a default subdirectory, allowed this to be altered, and then got on with things.

Scanning

As of 12 April 1996, *AVSK* claims knowledge of 8420 viruses. For reasons I could not sort out, the DOS version refused to access the virus test-sets stored on a magneto-



AVSK's general options allow for the depth of the scan to be altered at the user's discretion.

optical disk. Nothing I did could persuade it otherwise. This caused much file copying when the polymorphic test-sets were encountered. For reasons which are also beyond me, the *Windows* version was quite happy to access the drive.

The *Windows* version looks radically different from its DOS sibling, and proved very simple to use, with half a dozen on-screen buttons providing easy access to the main functions.

When used via drop-down menus, *AVSK* first scans memory, then displays the current subdirectory and its contents (in separate windows) – vaguely reminiscent of *CPAV*. Both DOS and *Windows* versions of *AVSK* offer options which can scan the entire system, an individual drive, a directory, a file, or the 'boot system'.

Scanning Speed

In its default state, the DOS version of *AVSK* scanned the hard disk of my test PC in 2 minutes 34 seconds (742 files in total, 311 files scanned, 29.9 MB).

AVSK recognises three types of compressed files (ZIP, ARJ and LZEXE). The option to scan inside compressed files is switched on by default, which of course slows down the scan. When this was deactivated, the hard disk of my test PC was scanned in 1 minute 40 seconds. With 'minimum stealth' specified, scan time fell again, to 1 minute 32 seconds. In the other direction, a scan of all files (including the contents of all compressed files) took 5 minutes 7 seconds.

Other methods of virus detection are included, and are even faster than the scanning itself. When *AVSK* searches for 'Suspicious Conditions', it inspects the entire hard disk in 37 seconds. A 'heuristic' scan takes just 2 minutes 40 seconds.

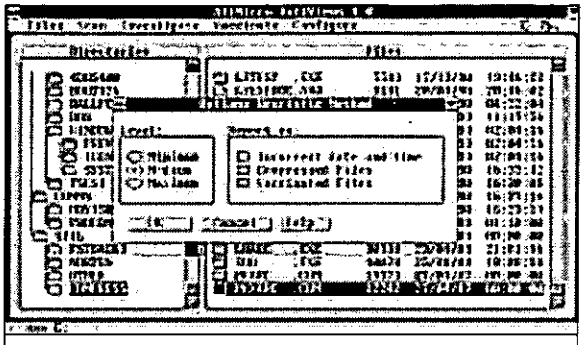
All the above timings were measured using the DOS version of the product. To provide a fair comparison, *Dr Solomon's AVTK* scanned the hard disk of my test PC in 4 minutes 21 seconds; *Sophos SWEEP* in 7 minutes 32 seconds.

One fact stands out, therefore: *AVSK* is very quick at scanning for viruses. A point worthy of note, however, is the slow-down in the other two scanners (*SWEEP* and *Dr Solomon's*) when SENTINEL is active: *SWEEP* took 13 minutes 6 seconds; *Dr Solomon's*, 15 minutes 1 second. This slow-down, imposed by the presence of SENTINEL, is severe.

Detection

I tested the virus detection capability of *AVSK* against the test-set described in the Technical Details section below.

Run against the In the Wild test-set, using default settings it detected 192 of the 286 test samples; 67%: frankly, not good enough. Curiously, the report file stated that *AVSK* had found 200 viruses, though only 192 files were infected: this was because all six samples of Jerusalem.1244 were detected as infected with (using *AVSK* nomenclature) both the Maca and the 1244 viruses, and the two samples of Keypress.1232.A were detected as doubly infected (Keypress and SamSoft).



The product showed no significant difference in detection rates despite the method of heuristic scanning chosen.

Against the Standard test-set, again using default settings, *AVSK* detected 229 of the 265 test samples (86%). Once again, samples (four in total) were detected as doubly infected (Warrier, 2 x Old_Yankee, Vienna). The DOS and *Windows* versions of *AVSK* detected the same viruses from the two main test-sets.

If the Standard and In the Wild test-sets are contained inside a ZIPped archive file, detection is slightly poorer. Only 180 files from the In the Wild test-set (63%) were found infected, and only 224 files from the Standard test-set (84%). I shall return to scanning inside archive files below.

Executing the 'Suspicious Conditions' option, rather than merely scanning, found 62 suspicious files in the In the Wild test-set (22%), and 51 (19%) in the Standard test-set. The heuristic level can be set to Minimum, Medium, or Maximum, although I could not measure a significant difference in detection when this parameter was varied. It was, however, ironic to find that 'Maximum Heuristics' found just one suspicious file: *AVSK's* own file SHIELD.COM. This was called an 'Unconventional Resident Program'. Ah well!

Polymorphic Viruses

When run against the polymorphic virus samples, *AVSK* detected 2196 of the 5500 test samples, or 40%. Three polymorphic viruses are detected only reasonably well (Girafe:TPE, Neuroquila.A and One_Half.3544), but the others are either not detected at all (three in total) or only very poorly (the remaining five).

When *AVSK* scans inside ZIP files, polymorphic detection falls off alarmingly – only 1209 (22%) of the samples are detected as infected. All but four viruses are completely undetected, and only Girafe:TPE is detected reliably. I am at a loss to see why this should be so. Surely, once a file has been extracted from a decompressed archive file, the same scanner should be used to test whether or not an infection is present? Clearly something is wrong with *AVSK's* decompression.

The product detected only fourteen of the twenty boot sector test samples, failing to detect EXEBug.A, IntAA, Peanut, Quox, She_Has and Urkel. By no stretch of the imagination can this be called an impressive result.

Vaccination

AVSK can create a database of checksum information about each executable file present on a hard disk (a process which it calls external vaccination), or it can add extra code to executable files (called internal vaccination). I am amazed that a manufacturer still considers changing executable files: use of such features is not recommended. Life is complicated enough without having to track down the inevitable problems that tampering with executable files may cause.

The product puts the two files which comprise the database for 'external' vaccination in the hard disk's root directory. I wish programs wouldn't do this. I am happy for *AVSK* to maintain a database, but it should do so in its own subdirectory.

When either creating or verifying external vaccination, the DOS version of *AVSK* took 3 minutes 50 seconds to work its way through the entire hard disk of my test PC, rising to 6 minutes 10 seconds under the *Windows* version.

Memory-resident Software

AVSK contains two distinct memory-resident anti-virus programs. One (SENTINEL) is a memory-resident scanner, the other (SHIELD) is a behaviour blocker. The documentation calls SHIELD a 'memory-resident program, whose mission is to prevent the damage that a known or unknown virus may create...': what this means is that it monitors (and prevents) certain actions; e.g. it can be set up so that any write to hard disk only takes place after user confirmation has been given.

SENTINEL can be added to AUTOEXEC.BAT by the installation program (see above), but SHIELD must be invoked by the user (either manually or as an addition to AUTOEXEC.BAT). When installed, SENTINEL uses 18.8 KB of conventional memory and 32 KB of expanded memory. SHIELD is much smaller, requiring only 3 KB.

When SENTINEL was executed with the /AE switch to ensure that all file extensions were scanned, my test PC locked up, complaining it could not load COMMAND.COM.

Testing Files before Execution

Any memory-resident monitoring program which carries out tests before allowing a file to be executed must have an impact on system performance. I measured this by copying 40 files (1.25 MB) from one subdirectory to another. With no memory-resident software present, it took 21.6 seconds, rising to only 22.1 seconds when SENTINEL was present.

This is very impressive. The result moved, however, to inducing curiosity when the file copying time went *down* to 21.3 seconds, with SHIELD added to SENTINEL.

Given the lack of overhead introduced by SENTINEL and SHIELD, it is difficult to explain why SENTINEL had such a drastic effect on the speed at which other scanners execute. Something odd is going on.

Behaviour Blocking

It is only necessary to use a PC with SHIELD active in memory for a few minutes to realise why the developers separated the two memory-resident programs. Put bluntly, SHIELD is a nuisance. If activated with all security options active, it is forever popping up and requesting confirmation. If some of its security features are turned off to prevent such intrusions (a hot-key is provided to facilitate such tailoring), effectively, SHIELD is doing nothing.

SHIELD is not alone in being intrusive or useless – all behaviour blockers tend to be like this. As a virus is merely a computer program, there is no foolproof way to decide which actions to allow and which to prevent. The only solution is to keep asking the user for confirmation as to whether a certain action should be permitted: this fails, as the average user has no idea how to answer such questions.

Therefore, although at first sight behaviour-blockers seem like a good idea, they come off the rails when the real world intrudes. SHIELD may have some use in constrained environments where users are to be allowed only a few actions, although I'm unconvinced.

Memory-resident Detection

When SENTINEL is executed, it states that it looks for only 420 viruses. Detection capabilities were measured by copying the files in the In the Wild and Standard test-sets from one drive to another. SENTINEL detected 179 and 187 infected files respectively in each set. These figures are only slightly less than the main *AVSK* scanner; surprising, given the low number of viruses about which SENTINEL claims knowledge.

Conclusions

My conclusions about *AVSK* are simple: it is very quick at scanning for viruses and/or verifying that checksums are unchanged, but simply not very good at detecting viruses. The memory-resident scanner is similarly poor, although surprisingly close to the DOS product in terms of detection. However, the behaviour-blocking memory-resident component is just plain annoying. Avoid it.

Technical Details

Product: *Anti-Virus Survival Kit v4.0* (no serial number visible).
Developer/Vendor: *AllMicro*, 18820 US Hwy 19 N, Suite 215, Clearwater FL, USA. Tel +1 813 539 7283, fax +1 813 531 0200, BBS +1 813 535 9042, email allmicro@ix.netcom.com.
Availability: *IBM PC* or *PS/2* or compatible running *MS-DOS v3.3* or higher with at least 512 KB of RAM. *Windows* components require *Windows 3.x* or higher with at least 2 MB of RAM.
Price: The base package can be downloaded from the company's Web site (http://www.allmicro.com/). Twelve months of updates cost US$79.95; six months, US$39.95.
Hardware used: *Toshiba 3100SX*: a 16 MHz 386 laptop with a 3.5-inch (1.4 MB) floppy disk drive, 40 MB hard disk and 5 MB of RAM, running under *MS-DOS v5.00* and *Windows v3.1*.
Viruses used for testing purposes: Boot sector test-set – see *VB*, March 1996, p.23. Standard, Polymorphic, and In the Wild test-sets – see *VB*, January 1996, p.20.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YP, England

Tel: 01235 555139, International Tel: +44 1235 555139
Fax: 01235 531889, International Fax: +44 1235 531889
Email: editorial@virusbtn.com
CompuServe address: 100070,1340
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield, CT 06877, USA

Tel +1 203 431 8720, fax +1 203 431 8165

END NOTES AND NEWS

***Sophos Plc's* next anti-virus workshops** will be on 25/26 September 1996 at the training suite in Abingdon, UK. The two-day seminar costs £595 + VAT. One single day may be attended at a cost of £325 + VAT (day one: Introduction to Computer Viruses; day two: Advanced Computer Viruses). For further information, contact Julia Line on Tel +44 1235 544028, or visit the Web site; http://www.sophos.com/.

The *NCSA* is hosting the ***Web, Internet Security and Firewall Conference***, which will be held in San José, California from 30 September to 1 October. Details on the event can be obtained from the *NCSA*; Tel +1 717 258 1816, fax +1 717 243 8642, or email fwcon96west@ncsa.com. Information is also available from their WWW site; http://www.ncsa.com/fw96west.html.

***Intel Corp* has announced the launch of *LANDesk Virus Protect for NT*.** The product offers on-access scanning using server-based technology. Information on obtaining the product is available from the company; Tel +44 1793 403000 (UK); +1 408 765 8080 (USA).

In Dorset, UK, a man has been charged with blackmailing *Sun Alliance*: it is alleged that Keith Lamb, from Bournemouth, **threatened to infect the company's computers with 'computer bombs and polymorphic codes'** [*Oooh! Ed.*] if a claim he had made (which had been rejected) was not paid. Lamb was arrested after his calls were recorded. A verdict in the case, being held at the Old Bailey, is imminent.

Mike Hill, director of product marketing at *S&S International*, claims that **a weakness in *Netscape* leaves it vulnerable to virus attack.** Hill asserts, according to an article in the UK publication *Computer Weekly* (18 July 1996), that if the right mouse key is used to activate the shortcut menu, all calls to add-in software are bypassed, leaving files unchecked. Because of this, *S&S* has now delayed the release of its *WebGuard*, which was designed to work in tandem with *Netscape 2.0*.

Integralis has announced a working partnership with S&S: **it will license its email and scanning technology** to the anti-virus software developer, which will be marketed under the name *MailGuard*. Information on the deal is available from *Integralis*; Tel +44 1734 306060, or on the WWW; http://www.integralis.com/.

Reflex Magnetics has another **Live Virus Experience** scheduled for 9/10 October 1996. Further information is available from Rae Sutton: Tel +44 171 372 6666, fax +44 171 372 2507.

Seven Locks Software has been named as the **exclusive US distributor for Czech-based *Alwil Software's* security products.** *Alwil's AVAST!* has been a prominent front-runner in recent *VB* comparative reviews. Details on the agreement are available directly from *Seven Locks*; Tel +1 508 746 9087, or on the WWW; http://www.sevenlocks.com/.

The *NCSA* has announced the first **certification of products in its Firewall scheme.** Sixteen products have so far met the criteria for acceptance: information on the procedures involved, and the products registered, is posted on the *NCSA's* Web pages; http://www.ncsa.com/.

***CompSec 96* will take place in London, UK,** from 23-25 October 1996. For information, contact Sharron Elmsley at *Elsevier Science*; Tel +44 1865 843721, fax +44 1865 843 9458.

S&S International is presenting **Live Virus Workshops** at the *Hilton National* in Milton Keynes, Buckinghamshire, UK on 2/3 September and 7/8 October 1996. Details are available from the company: Tel +44 1296 318700, fax +44 1296 318777.

Readers are reminded that the ***6th Annual Virus Bulletin Conference and Exhibition*** takes place in **Brighton, UK, on 19/20 September 1996.** Contact Alie Hothersall for information; Tel +44 1235 555139, fax +44 1235 531889, WWW; http://www.virusbtn.com/.

Virus Writers: The End of The Innocence?

Sarah Gordon

IBM Thomas J. Watson Research Center

sgordon@watson.ibm.com

Abstract

Earlier research has empirically demonstrated the cyclic nature of virus writing activity: as virus writers "age out", new virus writers take their places. Enhanced connectivity amplifies the existing problem and various technical factors result in new types of virus writers surfacing as the cycle repeats.

However, a new variable has recently been introduced into the cycle: high profile legal intervention. The virus writing community now has experienced visits by concerned law enforcement personnel; there have been arrests and there will be sentencings. New laws are being considered, enacted, and acted upon. Thus, the virus writing scene is no longer a casual pastime of kids on local Bulletin Board Systems.

What has been the impact, perceptually and operationally, of these visits, arrests, and sentencings? In other words, as the virus problem gets more and more "real world" attention, where are we actually going in terms of shaping acceptable behavior in our virtual communities and what, if any, effect are these legal interventions having on the impact of viruses upon users' computers?

In order to produce a scientifically meaningful answer to this question, pre and post intervention data on various aspects of the virus problem have been gathered. We solicited opinions on a variety of topics related to computer viruses and legal countermeasures via e-mail and direct survey. Opinions are not only interesting; they must be considered, as we know the opinions of today shape how people behave in the future. However, we are also concerned with immediate real-world impact. To this end, impact will be examined in terms of viruses found both In the Wild[1] (ItW) and on the World Wide Web (WWW), as a function of time. The data gathered before and after various types of high profile intervention is considered; in particular we are interested in any decrease noted in the graph of virus growth both ItW and on the WWW, and in online references to legal concerns.

An analysis of the data is presented and suggestions for future research are made.

[1] Using The WildList (http://www.wildlist.org)

Introduction

During the last eight years, a wealth of information has been gathered concerning virus writers and the various motivations behind their work (Gordon, 1994a; Gordon, 1994b; Gordon, 1995; Gordon, 1996; Gordon, 1999). In this paper, that earlier research is expanded upon and updated to consider an increasingly important facet: intervention by legal/government bodies.

It is natural, given the way societies tend to develop, that antisocial activities tend to lead to legislation designed to contain or eradicate the activities. This paradigm of control is influencing both technological development and societal direction (Gordon, 1994b). There is now increased pressure on the legislature and law enforcement to deal with a problem which purportedly costs corporations millions of dollars per year (Cobb, 1998). The goal of this paper is to gain insight into the efficacy of high-profile legal countermeasures, and assess how well they achieve the objective of lessening the spread of computer viruses.

In order to accomplish this analysis, this paper is structured as follows: First, the research to date is summarized, in order to provide the reader with insight on the "generic" virus writer, the target of laws and intervention. Second, the legal countermeasures which are in place at the time of writing are discussed, outlining the goal of legislation, and summarizing the laws employed in past high-profile arrests of virus writers. Next, the potential drawbacks and costs associated with this approach are discussed, to provide a counterpoint to the intuitively obvious application of laws and high profile interventions as a solution to the "problems" of virus writing. The lack of useful metrics as to the effectiveness of the legal approach is covered, before discussing a research methodology that provides scientifically valid data for assessing the result of the interventions. Finally, results of this research are presented, analysing the effectiveness of laws in the prevention of virus writing and various forms of distribution.

Virus Writer Demographics

Research published by (Gordon, 1994a) examined the demographics of a large number of virus writers. This was accomplished by the use of surveys, email interviews, online chat and in-person sessions. The data gathered was used to assess the ethical development[2] of individual virus writers, with a view to understanding why they chose to write viruses, and what, if anything, was likely to deter them.

The paper focused on four primary groups of people: the adolescent virus writer, the college student, the adult virus writer, and the ex-virus writer. The findings for each group are summarized below[3].

The Adolescent

Studies of the adolescent virus writer were remarkably consistent. The data tend to show that the adolescent virus writer is ethically normal and of average/above average intelligence. Responses from members of this group showed respect for their parents and for authority (to some degree). While members of the group tended to understand the difference between what is right and wrong, (i.e. directly damaging data that belongs to other people is wrong) they typically did not accept any responsibility for problems caused when their own viruses appeared in the wild.

The College Student

Members of this group also appeared to be ethically normal on the Kohlberg scale. Despite expressing that what is illegal is "wrong", members of this group were not typically concerned about the results of their actions related to their virus writing.

---

[2] based upon the Kohlberg model (Kohlberg, 1981; Panzl & McMahon, 1989)

[3] other models produced similar results

The Adult

Of the four classes studied, the adult virus writer was the smallest, and the only one which appeared to be ethically abnormal, appearing below the level of ethical maturity which would be considered normal on the Kohlberg scale.

The ex-virus writer

Once again, this group was ethically normal. The ex-virus writers typically cited lack of time and boredom with virus writing as the primary motivator for the cessation of their "hobby". Appearing socially well adjusted, the ex-virus writer seemed to bear no ill-will toward other virus writers, and was undecided concerning the ethical legitimacy of virus writing.

These results are of particular relevance to the question of legal countermeasures. The virus writing adults in the study appeared to be below the norms in ethical development; adults who are below these norms are more likely to be motivated by fear of punishment than by respect for law. For the adult virus writer, therefore, it is not the laws that are important, but their perception of the likelihood of being prosecuted under those laws. For the minors involved, the presence of laws is unlikely to be very effective for several different reasons that will be discussed in more detail later. For the youngest virus writers, it tended to show that virus writing was a naturally self-limiting phenomenon, and that the "perpetrator" would tend to cease their activity without the need for legal intervention.

The research shown above was completed in 1994. The update of the paper two years later (Gordon, 1996) showed some disturbing trends related to virus writers at the higher age limits considered. Whereas virus writers were typically aging out as their ethical development continued, mixed messages from many different sources appeared to make virus writing appear "less wrong", pushing up the age of aging out, if the process occurred at all.

Legal and High Profile Intervention

According to (ICSA, 1999) the median cost of virus disasters is $1,750, with some respondents reporting costs of up to $100,000 in a single virus incident. Another study (Ernst, 1998 cited in Cobb, 1998) suggests that virtually every organization in the world has experienced at least one virus infection, and that viruses continue to cause businesses hundreds of millions of dollars each year in damages and lost productivity. Given the purported high cost[4] to businesses it is not surprising that some people have looked to the law for help in dealing with the problem.

Legal intervention in the case of the Melissa virus has been highly publicized. Regarding this case, (Jenislawski, 1999) citing ICSA, states

> *"This case, the company says, proves that virus writing is 'indeed illegal', despite arguments to the contrary.* [This prosecution] *will be a decisive event that will tend to reduce the relentlessly increasing threat and resultant risk of computer viruses to society as a whole. By locking up perpetrators, the cycle of mounting numbers, rate, and virulence of computer viruses will get at least a pause and perhaps, a reversal. '"*

(Tippett, 2000), suggests that Congress look at making it illegal to write a computer virus. *"Making a bomb is illegal but writing about how to make a bomb is not"*, he noted. *"But with a computer virus, the words are the bomb"*. (Kabay, 2000a) calls for a view of computer programs as "not speech". [5]

---

[4] social effects related to lack of trust are outside the scope of this paper

[5] an in-depth discussion of viruses as speech is outside the scope of this paper

How effective are these legal counter-measures likely to be in addressing problem of viruses found in the real world? In (Lemos, 1999) we read

> *"Despite an expected four- to five-year sentence for admitted Melissa virus writer David L. Smith, the number of new viruses appearing on the Internet appears to be accelerating as the end of the millennium draws near, anti-virus firms said Friday."*[6]

Laws to combat computer crime are not new. The first comprehensive proposal for computer crime legislation was a federal Bill introduced in the US Congress by Senator Ribikoff in 1977. (Schjolberg , 2000). Since that time, many U.S. states have introduced various computer crime laws, several of which mention viruses specifically (Bordera, 1997).

Some of these laws and statutes even attempt to define what a virus is. For example (Bordera, 1997) cites the revision of the State of Maine's statute title 17-A, ßß 431 to 433 (West Supp. 1996)

> *"any instruction, information, data or program that degrades the performance of a computer resource; disables, damages or destroys a computer resource; or attaches itself to another computer resource and executes when the host computer program is executed."*

The State of Maine has a particular subsection dealing with viruses, ß433c, citing

> *"intentional or knowing introduction or allowing the introduction of a computer virus into any computer resource, having no reasonable ground to believe that the person has the right to do so."*

The offense is classified as a Class C crime.

In (Froehlich, Pinter, and Witmeyer, 2000) documentation of differentiation between naivete and malice is made:

> *"The 1994 Computer Abuse Act tries to deal differently with those who foolheartedly launch viral attacks and those who do so intending to wreak havoc. To do this, the Act defines two levels of prosecution for those who create viruses. For those who intentionally cause damage by transmitting a virus, the punishment can amount to ten years in federal prison, plus a fine. For those who transmit a virus with only "reckless disregard" to the damage it will cause, the maximum punishment stops at a fine and a year in prison."*

There have since been various committees formed worldwide that have attempted to deal with the problem from a legal perspective (Schjolberg, 2000). From some of these committees international laws addressing computer crime have emerged, some of which address virus issues specifically. For example, in 1995, the Iranian Government approved a computer crime law prepared by the High Council of Informatics. Program damage caused by viruses, Trojan horses, worms, and logic bombs are spelled out in this law. Other countries have laws that forbid the spreading of and in some cases the writing of, computer viruses (Iran, 2000). How have the existing laws been used so far? First, we will consider three individual cases.

Research by (Akdeniz & Yaman, 1996) documents the case of Dr. Joseph Popp, an American who was apprehended and arrested by the FBI at the end of 1989. Dr. Popp had sent free computer diskettes to ~20,000 people in London and around the world; these disks contained a program which supposedly assessed the user's risk of contracting the AIDS/HIV virus, but which in reality introduced a trojan horse to the users computer. According to Akdeniz,

> *"Recipients of the disk were warned that their computers would stop functioning unless they paid the license fees of £225 to a bank account in Panama. This case is thought to be the world's most ambitious computer crime. While Dr. Joseph Popp was extradited to the UK, his case never came to trial due to a deterioration of Popp's mental state; he was found mentally unfit to stand trial."*

---

[6] this assertion is examined later in this paper

(Taiwan, 1999) describes how, in 1999, the Computer Crime Unit traced the CIH virus to a young man then serving in the military. He confessed he had written the virus, claiming he was motivated by pure research, and had not himself spread the virus. According to this report,

> *"if it were determined that Chen Ying-hao had maliciously disseminated the virus, he could be sentenced to time in jail. However, many creators of computer viruses are computer jocks, most of whom write viruses to show off their computer ocumen. As Chen Ying-hao likely belongs to this ilk, and since under the article in question a prosecution can only be brought if a complaint is made, it has thus far not been possible to charge Chen, for lack of sufficient evidence. Prosecutors are currently reviewing the case."*

Christopher Pile, known as the "Black Baron" in the computer underground, was sentenced to 18 months on 15 November 1995. Pile was charged with violations of Section 3 of the Computer Misuse Act 1990. He pled guilty to five charges of gaining unauthorized access to computers, five of making unauthorized modifications and one of inciting others to spread the viruses he had written.

Laws – Effective?

In order for a crime involving a virus to be prosecuted, it must first be reported. Minnesota statute ßß 609.87 to .89 presents an amendment which clearly defines a destructive computer program, and which designates a maximum imprisonment or 10 years; however, no cases have been reported. Should we conclude there are no virus problems in Minnesota?

In (Grable, 1996) the ineffectiveness of the laws, both Federal and New York State, as a solution to the virus problem are clearly spelled out:

> *"Both the federal and New York state criminal statutes aimed at virus terror are ineffective because the methods of enforcement... The combination of the lack of reporting plus the inherent difficulties in apprehending virus creators leads to the present situation: unseen and unpunished virus originators doing their damage unencumbered and unafraid. Add to that the slap on the wrist afforded to even the most infamous of virus propagators, and the recipe is right for even greater damage from malevolent software."*

How likely are laws to affect the young virus writer? We first examine legal intervention related to young people engaged in other antisocial activities.

(McDowall & Loftin, 2000) analyze the success of curfew laws in controlling crime. They state that while several police departments report a decrease in youth offenses after the enforcement of curfew ordinances (Bilchik, 1996) claim that statistics supporting the efficacy of curfew laws in reducing crime rest on uncertain comparison groups, and that few evaluations have considered more than a single area. They conclude there is *not* strong evidence that the curfew laws reduce juvenile offending or victimization rates. However, despite this lack of evidence, these laws have been embraced by many communities; (Hemmens & Bennett, 1999) state that while it is unclear whether they are effective in reducing crime, it is clear that they are being embraced by communities across the country (Davidson, 1997).

In other studies of youths living in areas where anti-social activity is normal, some youth may accept confronting danger and being involved in these activities as features of living in such environments (Halliday & Graham, 2000). There is insufficient data to conclude if this phenomenon maps to virtual environments.

Research by (Foglia, 1997) supports the hypothesis that while the possibility police involvement, or legal sanction does not offer significant deterrence for youths who engage in antisocial behaviours, they *are* likely to be influenced by parents and peers. In (Gordon, 1994a), the conclusion that the "common" young virus writer is not likely to be affected by laws is supported, citing both the non-universality of the laws as well the mixed messages sent societally to the young people as they integrate into the cyber-culture.

Difficulty in sentencing minors is also to be considered; some research is being done in this area. (Simpson, 1999) examines research into state statutes in the United States that help make parents legally responsible for personal injury or damage to property made by their minor children. There are details on a case in Minnesota (the land of no viruses ☺), and another in Oregon, where such provisions currently exist.

Finally, we must *not* ignore the mixed messages sent to young people regarding virus writing. (ZiffDavis, 1999) reports

> "[the firm that hired the virus author]*...competed with a score of high-tech rivals attempting to lure* [the virus author]..."
>
> *"'Our chairman felt he* [the virus author] *was a rare computer professional and we decided to uccept him with an open heart,' said Wahoo spokeswoman Vivi Wang."*

Contrast that to the alleged writer of the Melissa virus, David L. Smith. Apprehended at the beginning of April, Smith is looking at a maximum sentence of 40 years if convicted in New Jersey State Court. The immense differences in punishment illustrate a large rift in perceptions over the seriousness of computer viruses.

Lack of Metrics

Perhaps one of the reasons that there are so many different opinions on the effectiveness of legislation is that little quantitative data has been gathered. How does one go about measuring the effectiveness of a law? While it is tempting to simply measure the number of arrests as a function of time and law, this is not a good approach given the small number of virus writers who have been arrested and tried. Indeed, this lack of arrests is one of the primary indicators used by some to argue that laws are not a good deterrent.

One of the ways in which we can judge the efficacy of law as a deterrent is the overall view of society toward the acts which have been criminalized (Bagaric, 1999). However, we must be careful not to impose our view of the act on others when attempting to use the criminalization as a "proof" that the act is "wrong". For example, the use of marijuana is a criminal offense in some places/situations; in others, it is a misdemeanor, and in yet others, it is an acceptable act.

New Metrics and Research Techniques

As virus writing is a relatively infrequent "crime", a better measure of efficacy might be to study the number of times this "crime" has resulted in viruses let loose into the user community. However, how shall we define this output of "crime"? While it is true that in practical terms, a measure of the virus problem can be derived from the infection rate per 1000 PCs, this figure is affected by far more than just the number or activity of virus writers. New types of virus, a virus "getting lucky", or simply press coverage for a well-known virus can skew this number. Similarly, the total number of known viruses is not necessarily a good indicator, as this number is somewhat artificial in its creation. Thus, we propose the following new metrics for measuring, albeit indirectly, the efficacy of legislation with respect to the virus "problem".

One possible way of measuring the prophylactic effect of laws is obvious: ask! Based upon previous research, we have built a reliable and open dialogue with many of today's more visible virus writers.

As this "known" population is relatively small (but has a large impact on many developments in the virus world) a directed survey was created and administered. Questions (shown in the results section) were initially distributed via electronic mail and in-person sessions to virus writers in North and South America, Asia, Europe and Australia. The questionnaire was also posted to the Usenet News Group alt.comp.virus. The theory is that by re-administering the questionnaire after a high-profile criminal case concerning viruses, any suppression in the tendency to write viruses could be documented.

Unfortunately, the sentencing of David Smith has been delayed several times, so at this time the administration of the post-test questions and analysis of that data is not possible. Following the sentencing of David Smith, the post-test will be administered and the results posted on the online version of this paper[7]. One drawback with this approach is that we expect some virus writers to become more socially aware as they "age out"; thus a significant delay between administering the two tests could make the results difficult to interpret for individual subjects. However, the average population should remain reasonably static, making the test a possible metric for evaluation of effectiveness of laws.

As intimated above, the full measure of the scope of the virus "problem" itself is extremely hard to measure. How "bad" is the "problem"? Can it be measured by the number of known viruses on a particular date? The number of viruses encountered "In the Wild"? The infection rate per 1000 PCs?

The answer to this question depends partly on perspective and partly on the need for the measurement. For example, from the perspective of the anti-virus researcher working in a non-automated environment, the scope of the problem is probably based upon the sheer number of viruses, as he must deal daily with all incoming virus, analyzing, meticulously naming and prioritizing them, creating cures, etc. For the researcher in an automated environment, the measurement is likely to be those viruses which cannot be handled automatically and which she must deal with manually. For the end user, the infection rate per 1000 PCs in environments which are representative of his or her own is a vital statistic. However, from the perspective of the legislator, the scope of the problem is probably related to the sheer number of problematic viruses - viruses which are highly publicized and brought to his attention - as this is a direct measure of the number of "illegal" or "undesirable" acts occurring (not allowing for natural corruption of existing viruses etc[8]).

As it seems unlikely that *writing* a virus that never ever is distributed would be made illegal in The United States, we propose that a suitable measure of the problem for a legislator is the number of viruses found "in the wild". Thus, it might be interesting to correlate the rate of change of the number of new viruses in the wild with high-profile prosecutions of virus writers. To this end, we have charted viruses "in the wild" as a function of time. If a noticeable decrease in the number of new ITW viruses is observed following an arrest/sentencing, the case could be made that the trials were helping the overall computer user population.

Another metric for the efficacy of laws is the availability of viruses on the WWW. We performed an in-depth analysis using one popular search engine, with the keyword of "virii", as a way of locating web sites that appeared to have content bearing further analysis. Once again, if the number of "virus exchange" web sites (sites containing live viruses or viral source code) could be shown to decrease with new legislation/prosecution, there would be evidence for the effectiveness of the current legislative attempts at controlling the spread of computer viruses.

Finally, there is the question of a possible backlash against legislation outlawing the development and distribution of computer viruses. As tracing a virus author is extremely difficult *if* the virus writer takes adequate precautions against a possible investigation, there is a possibility of a backlash against any legislation which a person or group deems unconstitutional or as an infringement.[9]

---

[7] http://www.av.ibm.com, http://www.badguys.org

[8] Liabilities and legislation related to naturally occurring software or hardware induced corruptions are beyond the scope of this paper.

[9] further discussion on cyber-activism or civil disobedience is outside the scope of this paper

To this end, a survey was conducted at the 2000 DEFCON conference held in Las Vegas. The conference, attended by many "white hat and black hat hackers" represents an important part of the computer security "counter culture", and in many ways attracts the exact group that laws against virus writing would be aimed at. We selected people randomly as they entered the conference foyer[10]. To help ensure people could understand the survey questions, and answer coherently, the selection was done on the first day of the Conference, early in the day, in order to sample people before they were intoxicated.

Results

The results from direct interviews provide an entirely subjective (but collectively representative) view of how people said they felt about the following four questions:

1. What (if any) impact do you believe the arrest of David Smith has had on virus writing and virus distribution to date?
2. What (if any) do you believe is a fair and just sentence for David Smith?
3. What do you believe his sentence will actually be?
4. What (if any) impact do you think the sentencing of David Smith will have on virus writing and virus distribution post-facto?

We shall now consider each question in turn, and show data from several differently classified sources.

The Impact of the Arrest of Smith

The following results are broken down into those involved in the virus writing/virus exchange scene, and those who are not (primarily, but not exclusively, virus researchers)

Virus writers and exchangers:

> *"I'm not sure I've seen any change in virus distribution. There's as little interesting code being released as there was, and as much crap as ever. More to the point, those who are clueful knew that someone was going to be 'tracked down' and 'busted' soon. Those who are clueful aren't releasing code anyway (at least, not to the public). Those who aren't clueful don't understand how David Smith got busted and are probably still doing what they were doing before Smith got busted.*
>
> *If anything, the effect was on virus writing. There were probably people out there who thought about writing viruses for fun, but got scared out of it for fear of 'getting busted'. I don't think we'll see it making a big impact on the quantity or quality of viruses out there--but it probably stopped a few kids from 'turning to the dark side'. :)"* (Anonymous, 2000a)

> *"His arrest has made some authors more cautious about handing out their work to just anybody, or even putting their name on it. However at the same time, it has outraged many other authors who are now using it as an excuse [and justification] to speak out about the ills of our society, and dare I say "justice" system.*
>
> *I'd say that overall it has balanced things out, and had no real long term effect in the minds of authors, it's only set a legal precedent."* (Anonymous, 2000b)

[10] 161 subjects, 90% confidence level, 6.0 confidence interval

On the writers side, none. Foul things can happen when you code such programs, and most writers know that already. The thought of a guy getting screwed by media hype is not going to stop most people from coding what they think is interesting.

The distribution side is a bit different. Alot has changed since the shitstorm (pardon me, but there is no nicer way to describe it) of april 99. The loss of the sourceofkaos server was a big deal to us. The vx scene hud a voice, and was stripped away due to the incident. The guy who hosted (we knew him as jtr) it was running the machine at his place of business. He was placed on paid leuve for a few weeks, and was let go. Im sure the fbi had a field day sorting through that box. Mediu, the av industry, government organizations would connect to the irc which didnt help much, due to kids that didnt really know the half of what was going on a spreading rumors and publicly discussing things that they shouldnt have. Ugh, it was a mess. Those were some stressful days. This has changed alot on the distribution side. People are afraid to release information. I was the first one to come forward and give the source of iworm.zippedfiles to the public because i had to. After the minimal heat it created, a handful of news articles and such on how the fbi was in search of its author, nobody (well, only a handful had the source in the first place) wanted to come forward with it. Posting source code is not breaking the law in most of the world. People should be afraid. (Anonymous, 2000c)

Antivirus researchers:

"It has had the impact that many very active virus writers have "retired" (seen anything from the Internal guy any time recently?), others have become less productive, and many have refrained from releasing their viruses into the wild. I think thut if Smith wusn't arrested so swiftly, we would have seen much more Melissa variants and many more from them would have been released into the wild in a similar fashion.

Of course, sooner or later this beneficial effect will weur off. People tend to forget, and young people, like most virus writers are, tend to forget even faster. That's why the law enforcement must not "sleep on their laures" (sic) *but must prosecute similarly swiftly offenders like Mr. Smith in the future, too."*(Bontchev, 2000)

*"I would hope that maybe it has scared away few would-be writers or discourage some from distributing their creations but I have seen no clear evidence of this. I'd say there would have to be at least *some* positive effect from this (I just don't have any evidence for that though.)"*(Stiller, 2000a)

"It did not have any and will not have any. Virus writer wrote, write and will go on writing viruses, whether one of them folks was, is or will be sentenced or not....None. We do not suw a change after Black Baron was arrested and I do not saw a decrease of new viruses..." (Marx, 2000a)

Two other responses are worth further examination. First, from the ever-scientific (and correct!) Mich Kabay (Kabay, 2000b)

"Don't know without research. What I hope is that it will discourage some of the virus writers, but that's pure conjecture."

The second sums up a practical point of view with good evidence behind it:

"Very minimal. Most virus writers (in my opinion) think that it was a fluke that he got caught. Very little, I thing that a one off situation will not change the ways of virus writers. Only if a lot of writers - distributors where caught would this make a impact." (Pineda, 2000).

Fair and just sentence for David Smith:

Virus writers had mixed opinions.

"Hard to call. I don't reully know the facts of the case. If he was maliciously distributing the code, I don't have much in the way of sympathy." (Anonymous, 2000d)

"An apology for ruining his life of future employment in the computer industry, a smile, and a handshake from every person that has cursed him. And perhaps a job. That's right". (Anonymous, 2000e)

"To be honest, I really haven't been following the David L Smith case. But I'd say upprox. 10 years max. As I once studied the law and jail sentances in an assignment about the meaning of life imprisonment (my best bit of school work that was) - and Life is only about 15-20 years. Computer data is for less important than human life, and should be judged accordingly" (Anonymous, 2000f)

"A slap on the wrist. Im not saying it was right to post a virus to a newsgroup from a stolen aol account. What he has already had to deal with should be enough though. I don't think anyone would go the same route twice. Being held at gunpoint and treated as a terrorist is a bit disturbing im sure. Jail time or fines wont help, nor will locking him away trying to set an example to others. Look at kevin mitnick, doing almost 5 years without a trial and denied bail hearings. Have people stopped or even cut back on cracking machines? Of course not." (Anonymous, 2000g)

Antivirus researchers expressed a variety of opinions:

"He certainly deserves substantial jail time and fines." (Stiller, 2000b)

"That's for the judges to decide. He has to be punished. Something like a year in prison and a BIG fine would do." (Gryaznov, 2000)

"I personally believe that David was stupid, ruther than malicious, and I therefore think the sentence should be similar to the one handed out to the author of the famous 'Internet Worm' (whatever that was - I'm not sure)" (Shipp, 2000b)

"... a suspended prison sentence (or time already served), some community service that will mean nothing to him, a fine he won't be able to pay, all resulting in an extremely high paying job in the field of computer security for an obscure consulting firm who will brag about their proven expertise in computer viruses." (Pichnarczyk, 2000)

What will the sentence will actually be.

Virus writers were uncertain; a typical response is shown here:

"It will probably begin by looking insanely harsh, and come out to something that is soft on prison time, and nasty for his future. Some of thut 'unable to be within 500 yards of a computer' bullshit, probably. "(Anonymous, 2000h)

Antivirus researchers opinions were diverse:

"Probably a small amount of jail time". (Stiller, 2000c)

"I think he will get a large fine, and 10 years." (Shipp, 2000)

"Some years arrest... maybe much too long, even if the virus cleun-up etc. costs very much." (Marx, 2000b)

"Suspended sentence, probation for a couple of years, specific interdiction of further computer-virus writing, and a fine of a few thousand dollars." (Kabay, 2000c)

What (if any) impact do you think the sentencing of David Smith will have on virus writing *and* virus distribution post-facto.

Virus writers were consistent within their grouping:

"None. It is the fear of being caught that is more important to an author, than the results that occur after. For example, even if this particular case was settled in David's favour, he would still be ruined in the computer industry. That's enough." (Anonymous, 2000i)

"None. Things like this only effect people when its in the spotlight. Its all said and done, its old news, the media wont rave about it, the end. It wont be forgotten, but it wont effect the future. Nothing changed from the black baron did it?" (Anonymous, 2000j)

Antivirus researchers:

"Marginals will stop. Hard-core will continue. After the Next One (tm) goes down, more will stop". (Thompson, 2000b)

"It depends upon the amount of media exposure and the severity of his sentence. I expect it would discourage some virus writers from distributing their creations." (Stiller, 2000d)

"Future arrests so as to make them commonplace will have such an effect. The precursor to that is "interest" from the authorities. As David Smith is responsible for creating the "interest," he will have had a tremendous impact on the future of such. But only if the authorities maintain the vigilance" (Kuo, 2000)

"An overly harsh sentence / treatment could make him into a martyr (cf. Kevin Mitnick). Too light a sentence would reduce the deterrent effect.

Overall, not a great deal, I strongly believe that the probability of getting caught is as important as the severity of the sentence in deterring potential criminals. For example, it is illegal to smoke in lifts (sorry, elevators in American translation) in HK, and lifts have signs saying the penalty is HK$5000. However, I often enter a lift and smell cigarette smoke, and I have never seen or heard of someone being fined. The chance of getting caught is (virtually) nil, so the heavy fine is no deterrent. If the fine was HK$100, but offenders were caught 50%+ of the time, the practice would quickly stop. Very few virus writers or distributors have been caught, so the severity of punishment is small deterrent." (Dyer, 2000)

"It's a mixed message. On the deterrent side, it's the classic "they'll think twice because they might go to jail" (if my desired sentence is carried out). On the flip side, it also shows virus writers how hard it is to prosecute & convict, as well as suggesting new methods for not getting caught. Ultimately, the impact will be low until the conviction volume increases." (Renert, 2000)

Survey Results and Analysis

This data shows an interesting cross section of views from both the anti-virus community and the Virus Writer/vX community. Interestingly, the vX community seems less convinced that laws will help the situation. This position does not appear to be based upon a vested interest in the unsuitability of laws, but a genuine feeling within the community that legislation will not be an effective preventative.

Perhaps the most cogent summary of this logic comes from (Dyer, 2000) quoted in response to Question 4, "Will the arrest and sentencing of David Smith have any long-term impact?": if the law will not be enforced or is unenforceable, it has little effect regardless of the penalties.

Table 1 shows a summary of the results from our survey:

	Yes	No	Maybe
Virus Writers			
Has the arrest of Smith had any impact in the virus writing community?	0	11	0
Will it have any long-term impact?	0	11	0
AntiVirus Researchers			
Has the arrest of Smith had any impact in the virus writing community?	8	7	1
Will it have any long-term impact?	7	6	3
***NB: Incidental comments include (1) too harsh sentences would be bad (2) more computer ethics classes would help and (1) requires more research**			

Table 1: Survey data. A questionnaire concerning the impact of the arrest of David Smith was presented to two different groups: those involved or in some way associated with virus writing, and those active in the anti-virus community. Note the strong reaction from the virus writers, who were emphatic that neither Smith's arrest nor any conviction/sentencing would influence them or the virus writing community in general.

Interestingly, the data is reasonably similar to a comparable survey conducted in (Briney, 2000). In the Briney survey, an informal poll was conducted among 25 well-known information security professionals, asking "will the sentencing of David Smith reduce virus writing". Of the 25 respondents, 11 said, "No", the Smith conviction will not deter others, while 9 said, "Maybe". Only 5 said "Yes".

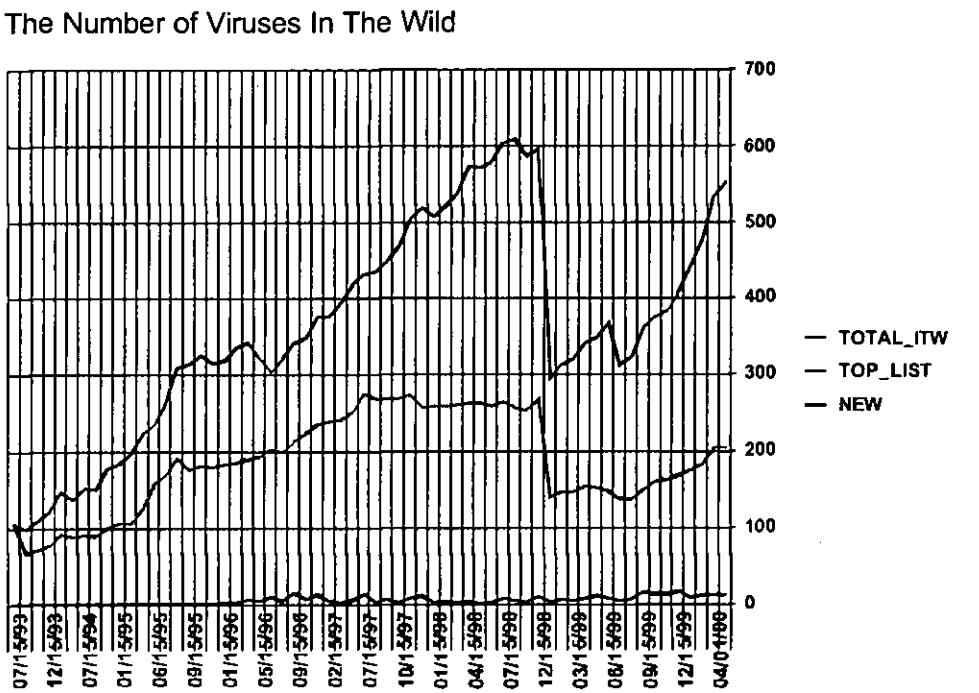


Figure 1: The Number of Viruses on the WildList as a function of time. This graph shows the number of viruses reported on the WildList as a function of time. The top (red) line shows the total number of viruses in the wild, the middle (green) line indicates just those viruses that are on the top portion of the WildList. Finally, the bottom (blue) line shows the number of new viruses added to the top part of the list per month.

As described above in the section *New Metrics and Research Techniques*, the total number of viruses In The Wild could be used as a metric of the efficacy of laws. In particular, we are interested in any discontinuity noted in the graph of viruses both newly ItW and also on the total number of viruses.

Before analysis can take place, the following descriptors should be made clear. The x-axis on the graph represents months of the WildList. The top (red) line represents the total number of viruses on the WildList, and the middle (green) line is those viruses reported by two or more reporters. Finally, the bottom (blue) line represents the rate of addition of new viruses per month. [Note that this information was only tracked from month January 1996, and so before this time the value is set to zero.]

The large discontinuity in the first two lines around January 1999 is an artifact of the change in methodology in the reporting structure of the Wildlist which resulted in a significant cleaning of the Wildlist data; rules concerning how long a virus must go unreported before being dropped from the list were enforced, leading to a significant drop in the total number of viruses listed. Note no corresponding discontinuity in the lower line; this is due to the fact that the corrections were not related to the rate of addition of new viruses, merely the renormalization of those already reported.

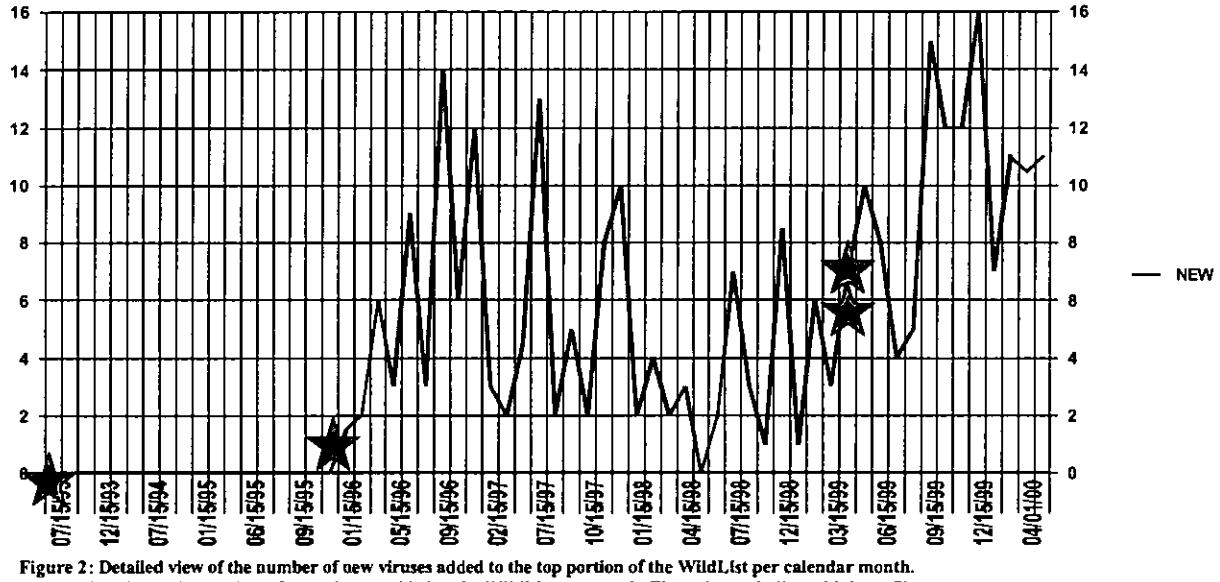


Figure 2: Detailed view of the number of new viruses added to the top portion of the WildList per calendar month. The red line shows the number of new viruses added to the WildList per month. The red stars indicate high-profile interventions. Note that there is no obvious drop in the rate of new viruses after these interventions.

As the most interesting, and arguably most relevant, data is the rate of new viruses becoming prevalent ItW, Figure 2 shows a detail of this data: On this graph, we have added stars to note prominent virus/trojan interventions or prosecutions[11]. As can be seen, the graph presents no clear evidence of any suppression in the rate new viruses were added to the Wildlist. While it can be argued that the data is (a) noisy (b) made up of more than one factor (that is, perhaps if there were no prosecutions, the graph would show a much-increased gradient) (c) lagging behind of real-world events due to the time it takes for a newly-released virus to spread and reporting cycles, one must also agree that the Wildlist data *provides no evidence to indicate that these high profile cases and*

[11] Popp, Pile, Ing-han, Smith

prosecutions have helped depress the virus problem as measured by the rute of addition of new viruses in the wild.

As this paper represents a snapshot of ongoing research and data gathering, not all the results have yet been gathered. One important metric proposed in the proceeding section was to measure the availability of computer viruses on the WWW. In order to do this, we measured the number of hits generated upon searching for the word "virii", using the Google™ search engine[12]. We examined each site to see if it offered viruses. The following results were noted:

On March 15, 2000 Google results netted 5080 for "virii". A manual examination of the first 1000 hits netted 65 sites with viruses (in executable or source code form) available for download. This means that approximately 6.5% of those sites surveyed contained live viruses or source code.

On August 18, 2000, Google results netted 20,600 results for "virii". An examination of the first 360 hits showed 102 sites with viruses (in executable or source form). This means that 28% of the sites surveyed contained viruses; a significant increase over the first data set.

It should be noted that the interesting figure in this experiment is **not** the total number of hits, but the percentage of those hits which contain viruses. As can be seen from the results, the percentage of sites which contain the word "virii" that also have live viruses has increased. While some optimization in search ordering may be responsible for this increase, this change in percentage is not likely to be due to a simple increase in the number of sites surveyed. Thus, this test does not show any convincing evidence for a decrease in the availability of computer viruses – if anything, viruses are more readily available now than ever before. After the sentencing of Smith, it will be interesting to note any effect on these figures.

One interesting by-product of the research was that some web authors noted that laws (or more correctly, fear of legal consequences) have certainly suppressed the dissemination of virus samples from some of the sites. Here are some examples of verbiage used on some of the sites:

Figure 3: Screen shot from a vX site on August 8, 1999

Figure 4: Screen shot from a vX site on January 1, 1999

[12] Google displays web sites based on page-rank. Thus, it retrieves pages based on the number of other pages which point to it. Therefore, the more highly visited pages are ranked first, with new pages being added as they become more popular

However, new sites have taken their places, including this one in The Netherlands, where such activity is illegal.

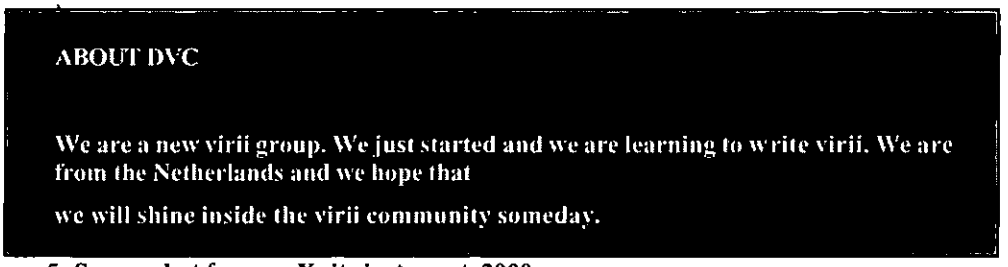


Figure 5: Screen shot from a vX site in August, 2000.

DEFCON Survey Data

A survey regarding reactions to proposed virus-writing legislation was also conducted. In this portion of the study, we chose the population of attendees at DEFCON (www.defcon.org), and asked two questions (The exact questionnaire is reproduced in Appendix A; however, the questions were posed verbally using the document as a reference):

- If virus writing were to be made illegal, would that make you less likely to write a virus (noted as Group 1); more likely to write a virus (noted as Group 3); or make no difference to your likelihood of writing a virus (noted as Group 2)?
- Given that what a person thinks is generally viewed as their own business, and that intentionally going out to cause someone problems with a virus by intentionally infecting their computer is viewed as "not ok", where on this scale of "how far would you go" do you personally draw the line at acceptable behaviour?

Then, we presented ordinally scaled actions ranging from those that would be almost universally accepted as right/okay, to an action that was almost universally accepted as wrong[13]. The resulting data is presented below as a set of histograms.

There are several different levels of analysis that can be performed on these data. At the simplest level, we can examine the data related to the first question: what was the stated effect of proposed laws. Interestingly, it seems that there is a significant set of people who claim that the criminalization of virus writing would encourage them to write computer viruses. Based upon verbal comments by the respondents, this was primarily due to their feeling that such a law would unfairly restrict their free speech.

Next, one can examine whether there is any correlation between the first answer and the second; that is, if we group the sample set based upon their reaction to laws, does one group appear more ethically developed than the other? Calculating the sample mean and standard deviation from each of the groups, we see that it is difficult to show any significant differences on the samples answers to question II based upon group. This is partly due to the fact that the data is clearly not normally distributed, although a visual analysis of the data does also tend to show a strong relation between the different groups.

[13] Time did not allow the preparation of a true Likert scale; this would be an interesting project for future research.

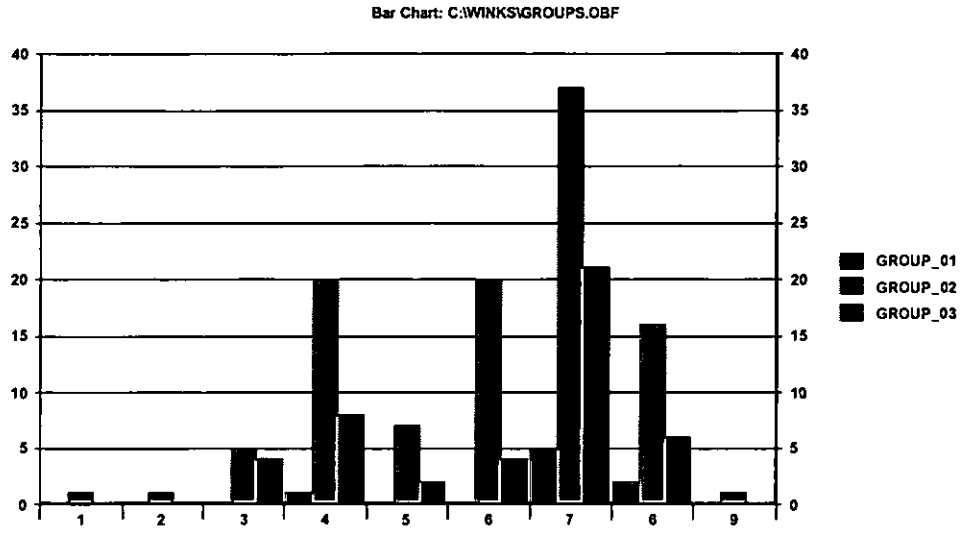


Figure 6: The effect of laws. Respondents were grouped depending on answer: those who would be deterred by laws (Group_01), those for whom laws made no difference (Group_02), and those who would be incited to write viruses by a new law (Group_03). Thus, new laws may cause an increase in the number of computer virus writers.

The fact that individuals with a low tolerance for virus exchange in general expressed that proposed legislation against virus *writing* would make it more likely they would write a virus is interesting.

It would be interesting to compare this data with that from students in a computer science course, in order to get some measure of the another population. However, in *ad hoc* studies conducted by the author within such environments, at least the reaction to proposed new laws appears to be similar.

Finally, it is interesting to note that some individuals mentioned that letting a virus you have written out of your own personal control accidentally was much more wrong than giving that virus to a friend; "stupidity" was cited as more wrong than intentional distribution.

Conclusions and Suggestions for Future Research

The focus of this research has been to gauge the impact of legal and high-profile intervention to the problem of damage caused by computer viruses. The data has shown that laws are of some limited effect in certain sections of the population, but that there could be a backlash in the United States to a law that was viewed to be a violation of an individual's rights to speech. While the free speech question as it pertains to computer viruses is unclear, this is immaterial: the key issue is that there are certain segments of the computing population within the United States who would *view* such a law to be unconstitutional, and state they would act accordingly. Further research on the likelihood of follow-through on electronic civil disobedience would appear to be an important next step in assessing the impact of legislation directly aimed at virus writing. Additionally, as the virus writing subculture is an international population, civil-disobedience and activism crossover between populations with laws and without laws bears further investigation.

A comparison of the number of viruses in the wild to high-profile virus writer cases/actions does not show any clear correlation with a decrease in the creation of new viruses. Indeed, despite much effort, the rate of addition of new viruses to the WildList appears to be increasing.

Tests and assessments should never be interpreted in isolation; thus, considering the strength of the responses can be as important in seeing the overall picture as the consideration of the statistical data. Additionally, this "strength of conviction" must be considered alongside the worldview of the

population. Consider that any laws created/enforced are aimed at a very small, but active virus writing community; the strength of conviction related to the DEFCON data seems to indicate that the creation of such laws would actually create more new virus writers than deter existing ones. This, coupled with the relative unenforceability of such laws could lead to a situation that is actually worse than the one we have currently.

Thus, examining all the data currently available, we are unable to show that the aggressive legislation directed toward, or intervention related to, virus *writers* will have any positive impact on the virus "problem" as defined by a number of different metrics.

We await the outcome of the post-sentencing interviews with interest. If the interviews show a significant change from their pre-sentencing results, proponents of thorough police follow-up of virus writers will have some hard data with which to back up their position. Conversely, if there is no appreciable difference in the data, we must, as a judiciary, re-evaluate the costs associated with pursuing legal remedies and high-profile "legal" interventions to a primarily sociological phenomenon.

Perhaps instead of attempting to raise support for making virus writing illegal, the energy and associated funds currently being expended would be better spent on education, with legal action or high profile intervention reserved for cases where an individual's clear and direct intent to damage could be shown.

An obvious objection to the lack of interventions is, quite simply, that the virus author should be held responsible for the results of his creation. After all, whether an infection occurs as the result of direct action from the virus writer (i.e. the virus is written, and uploaded to a Usenet News Group, masquerading as a legitimate utility) or is put into circulation via the WWW (i.e. clearly labeled as a virus on a virus exchange WWW site), the fact remains: someone created the virus that is responsible for the infection. The question is what, if any, responsibility does the creator of the virus hold?

In cases where a direct relationship between the virus author and a crime involving his virus can be shown, adequate existing legal measures can be applied. However, in cases where a virus author claims a "right" to make his or her virus freely available, or gives the virus away to knowing and willing recipients, but does *not* directly cause an infection, should we assume the question of responsibility dissipates? Opinions on the degree of responsibility vary, but one respondent's comments on this issue bear further examination:

> *"Shouldn't they really know by now that these things can cause problems whether they mean for them to or not!?"*

Unfortunately, in many cases we continue to see a typical pattern of older virus writers "aging out", while a new, inexperienced batch is still being birthed. By the time a virus writer is of age to know better, and to recognize the impact of these actions on others, they are already beginning to disassociate with their virus writing activities. Thus, while in some ways there is an "end of innocence" by those who realize their mistake, and exit the field, there is a complete pipeline of new authors just beginning their exploration. For this reason, it is flawed to simply assume that there is no innocent in the virus writing world; far from it: there are many.

This innocence and naivete, combined with the rapidly accelerating growth and evolution of technology, create a problem that is far more complex than socio-technological problems of the past. Other technologies that have been hugely influential on our societies have developed relatively slowly, thus enabling us to keep pace, predict future trends, and impart values related to those technologies to our young people. Now, however, the technology upon which we are attempting to base our projections is evolving rapidly. As the virus writing subculture continues to evolve, we are likely to see an exacerbation of problems relating to the technologies we are developing. The real question is how to best deploy our resources to protect us from this learning process, in which we are all participants.

Bibliography

Akdeniz & Yaman. 1996. *The Computer Misuse Act 1990: an Antidote for Computer Crime* First Published in Web Journal of Current Legal Issues in association with Blackstone Press Ltd.

Anonymous, 2000a-j. *Private e-mail correspondence*. Used with permission.

Bagaric, M. 1999. *Sentencing: The Road to Nowhere.* Volume 21 Number 4. December. The Sydney Law Review. University of Sydney, Australia.

Bilchik, S. 1996. *Curfew: An Answer to Juvenile Delinquency and Victimization?* OJJDP Juvenile Justice Bulletin .

Bontchev, V. 2000. *Private e-mail correspondence*. Used with permission.

Bordera, M. *The Computer Virus War: Is The Legal System Fighting or Surrendering?* Computers & the Law Project. Computers and Law, University of Buffalo School of Law.

Briney, A. 2000. *Private e-mail correspondence*. Used with permission.

Cobb, S. 1998. *Taming Wild Code.* Information Security Magazine. April.

Davidson, M. 1999. *Do you know where your children are?* Reason Online. November. http://www.reason.com/9911/fe.md.do.html

Dyer, A. 2000. *Private e-mail correspondence*. Used with permission.

Foglia, W. 1997. *Perceptual deterrence and the mediating effect of internalized norms among inner-city teenagers*. Journal of Research in Crime & Delinquency, Vol. 34 Issue 4, p. 414

Froehlich, J., Pinter, E. & Witmeyer, J. 2000. *Making The Time Fit The Crime.* Legal Column Archives. http://www.fmew.com

Gordon, S. 1994a. *The Generic Virus Writer.* From the Proceedings of the International Virus Bulletin Conference. Jersey, Channel Islands. pp.121 – 138

Gordon, S. 1994b. *Faces Behind the Masks.* Secure Computing Magazine. November 1994.

Gordon, S. 1995. *Technologically Enabled Crime: Shifting Paradigms for the Year 2000.* Computers and Security Journal. December 1995.

Gordon, S. 1996. *The Generic Virus Writer II.* From the Proceedings of the International Virus Bulletin Conference, 1996. Brighton, UK. pp. 177 – 188.

Gordon, S. 1999. *Viruses in the Information Age.* Virus Bulletin. June, July, & August. 1999. http://www.badguys.org/vb3part.htm

Gryaznov, D. 2000. *Private e-mail correspondence*. Used with permission.

Halliday, C. & Graham, S. 2000. Personality & Social Psychology Bulletin, May 2000, Vol. 26 Issue 5, p. 5480.

Hemmens, C. & Bennett, K. 1999. *Juvenile curfews and the courts: Judicial response to a not-so-new crime control strategy,* Crime & Delinquency, Jan99, Vol. 45 Issue 1, p99.

Grable, J. 1996. *Treating Smallpox with Leeches: Criminal Culpability of Virus Writers and Better Ways to Beat Them at Their Own Game.* Computers & the Law Project. University of Buffalo School of Law.

ICSA. 1999. *ICSA Releases 1999 Computer Virus Prevalence.* http://www.icsa.net/html/press_related/1998/virusprev98.shtml

Iran. 2000 http://www.gpg.com/homePages/peik/policies.htm

Jenislawski, S. *Melissa Virus Author Admits $80 Million in Damage.* http://www.policy.com/news/dbrief/dbriefarc439.asp

Kabay, M. 2000a. *Viruses are not Speech*. Virus Bulletin. "Comment" July 2000

Kabay, M. 2000b. *Private e-mail correspondence*. Used with permission.

Kabay, M. 2000c. *Private e-mail correspondence*. Used with permission.

Kohlberg, L. 1981. *The Meaning and Measurement of Moral Development*. Clark University Press. Worcester, MA.

Kuo, J. 2000 *Private e-mail correspondence*. Used with permission.

Lemos, R. 1999. *'Tis the Season for Computer Viruses.* http://www.zdnet.co.uk/news/1999/49/ns-12098.html. December.

Marx, A. 2000a. *Private e-mail correspondence*. Used with permission.

Marx, A. 2000b. *Private e-mail correspondence*. Used with permission.

McDowall, D. & Loftin, C. 2000. *The Impact of Youth Curfew Laws on Juvenile Crime Rates.* Crime & Delinquency, January 2000, Vol. 46 Issue 1, p.76.

Panzl, B. & McMahon, T. 1989. *Ethical Developmental Theory and Practices.* From the 71st Annual Meeting of the National Association of Student Personnel Administrators. Denver, Colorado.

Pichnarczyk, K. 2000 *Private e-mail correspondence*. Used with permission.

Pineda, R. 2000. *Private e-mail correspondence*. Used with permission.

Renert, C. 2000 *Private e-mail correspondence*. Used with permission.

Schjolberg, S. 2000. *The Legal Framework- Unauthorized access to Computer Systems.* Byrett, Norway.

Shipp, A. 2000a. *Private e-mail correspondence*. Used with permission.

Shipp, A. 2000b. *Private e-mail correspondence*. Used with permission.

Simpson, Michael. 1999. *Laws That Make Parents Pay.* National Education Association Today, Mar99, Vol. 17 Issue 6, p25.

Stiller, W. 2000a. *Private e-mail correspondence*. Used with permission.

Stiller, W. 2000b *Private e-mail correspondence*. Used with permission.

Stiller, W. 2000c *Private e-mail correspondence*. Used with permission.

Stiller, W. 2000d *Private e-mail correspondence*. Used with permission.

Thompson, R. 2000. *Private e-moil correspondence*. Used with permission.

Tippett, P. 2000. http://www.thesunnews.com/news/stories/2074548.htm

Taiwan, 1999. *Cuught in the Net. Is Cyberspoce a new haven for crimes.* Taiwan He@dlines. No. 70. http://www.taiwanheadlines.gov.tw/19991214/1999121413.htm

ZDNET, 1999. http://www.zdnet.co.uk/news/1999/51/ns-12354.html

Appendix A

These questions were presented verbally to a random sampling of attendees of the DEFCON conference.

Some people want the writing of self-replicating computer code to be illegal. If this were to become a reality, would you be:

(a) Less likely to write self-replicating code

(b) Not influenced one way or the other (makes no difference)

(c) More likely to write self-replicating code

Given that what a person thinks is generally viewed as their own business, and that intentionally going out to cause someone problems with a virus by intentionally infecting their computer is viewed as not ok, where on this scale of "how far would you go" do **you** personally draw the line at acceptable behaviour?

1. Thinking about writing the virus
2. Talking on a BBS about how you might write the virus
3. Writing the virus on your own computer, but never giving it to anyone.
4. Writing the virus on your own computer and having it escape accidentally
5. Writing the virus on your own computer and giving it to one or two friends
6. Writing the virus and uploading it to a VX site, labeled as a new virus.
7. Writing the virus and posting it to Usenet labeled as a useful application
8. Writing the virus and deliberately infecting other people's computers with it.

The Anti-Virus Strategy System

By Sarah Gordon
E-mail:sgordon@low-level.format.com

- Abstract
  1. Introduction
  2. Definitions
     - General Systems Theory
     - Holism
  3. Anti-Virus Strategy Systems
     - Components (with diagram)
     - Programs, Policy and Procedures (Selection, Implementation, Maintenance)
  4. Variations on a Theme
     - System Failure and Management
  5. Conclusion
- Bibliography
- About the Author

Abstract

Anti-virus protection is, or should be, an integral part of any Information Systems operation, be it personal or professional. However, our observation shows that the design of the actual anti-virus system, as well as its implementation and maintenance, can range from haphazard and sketchy to almost totally nonfunctional.

While systems theory in sociological disciplines has come under much attack, it has much to offer in the management of integration of technological applications into daily operations. We will examine the 'anti-virus' strategy (Policy, Procedure, Software [selection, implementation, maintenance]), focusing on areas where the 'system' can fail. We will address this interaction from a business, rather than a personal computing, point of view.

The Anti-Virus Strategy System will examine anti-virus strategies from a Holistic General Systems Theory perspective. By this, we mean that we will concern ourselves with the individual parts of the system, their functionality, and their interaction. We will draw from various IT models specifically designed to provide a holistic, forward-thinking approach to the problem, and show that for our strategy to flourish, we must concern ourselves with the system as a whole, not merely with its individual components.

Return to Top

Introduction

Computer virus. System failure. These words bring to mind a computer system brought to its knees - data corrupted and time wasted. Is this an accurate picture? We hear arguments against investing in virus protection: 'Viruses are mythical. Your chances of getting hit by one are pretty rare.' Others tell us anti-virus software is a necessity: 'Viruses can cost your company a lot of money. Better safe than sorry.' What are we to believe?

Let's assume that you don't have any anti-virus software. If you are 'hit' by a virus, the cost will be proportional to the value of your data and the value of your time. Independent studies [1] have shown that this cost can be quite high, depending on these factors as well as environmental factors such as how many computers you have (Note: If your data is of little or no value, and if your time is worthless, then you can well afford not to have an anti-virus strategy).

We will assume here that your data is worth something to your company, and that your time also has a significant value. In this case, you will want to protect your computer system from viruses. We will concede for the purists among us that not all viruses are intentionally harmful, but stipulate that intentional harm is not requisite for actual harm. For our purposes, allocating disk space and CPU time and/or modification of files without knowledge and consent (implied or otherwise) constitutes damage, as do deliberate or unintentional disruption of work, corruption of data and the lost time mentioned earlier. Basically, we are saying viruses are bad and we want to protect against them (there may be some wonderful new virus out there in development that can help us, but that is beyond the scope of this paper).

Fortunately, we are in luck. The very thing we need already exists: software, which will detect 100 percent of viruses listed by the Wildlist [2] as being known to be in the wild. In tests run against a library matched with the Wildlist, several programs were capable of detecting all such viruses. The necessity of detection of 'lab' viruses is another matter, and will not be covered at this time, although it is addressed in [3].

Since we have such software, we should have no problems. However, there are problems. Something is wrong. Before examining the sources of the problem, a few comments on definitions we will be using are in order.

Return to Top

Definitions

The definitions used here are pretty generic, and are adapted for use in an interdisciplinary approach to the problems addressed. Some among us would argue that the systems movement was born out of science's failures [4], but in this paper, we take the view that General System theory is a child of successful science, and as most children, it sees things through optimistic eyes. We have specifically avoided in-depth discussion of categorical schemes, generalizations, and other commonly used 'tools' of General Systems thought, and have focused instead on the simplest of the simple. The ideas in this paper are drawn heavily from very basic works in systems theory. They are not new ideas, but it is our hope that their application to the management of security and computer viruses will help us identify some of the problems we may be overlooking.

Return to Top

General Systems Theory

A system is a set, or group, of related elements existing in an environment and forming a whole. Systems can be made up of objects (computers), subjects (your employees) and concepts (language and communication); they can be made up of any one or more of these elements. There are 'real systems' (those which exist independent of an observer), and 'conceptual systems' (those which are symbolic constructs). Our system, 'The anti-virus strategy system', is not so different from many others, in that it is composed of all three elements: computers (objects), people (subjects) and concepts (policies and ideas). Each of these systems has its own subsystems. For example, your system of networked computers consists of individual computers. These computers are comprised of yet more subsystems; microprocessors, resistors, disk drives, etc. Our system consists of both real and conceptual subsystems. A system can also be said to be a way of looking at the world, or a point of view [5].

Concepts, laws, and models often appear in widely different fields [6] based upon totally different facts. This appears to be at least in part due to problems of organization, phenomena which cannot be resolved into local events, and dynamic interactions manifested in the difference of behaviour of parts when isolated or in higher configurations. The result is, of course, a system which is not understandable by investigating their respective parts in isolation. One reason these identical principles have been discovered in entirely different fields is because people are unaware of what those in other disciplines are doing. General Systems theory attempts to avoid this overlap in research efforts.

There are two main methodologies of General Systems research; the empirico-intuitive and the deductive theory. The first is empirical, drawing upon the things which regularly exist in a set of systems. It can be illustrated fairly easily, but lacks mathematical precision and can appear to the 'scientist' to be na‹ve. However, the main principles which have been offered by this method include differentiation, competition, closed and open systems, and wholeness - hardly na‹ve or worthless principles. The second method, basically, can be described as 'the machine with input', defined by a set 'S' of internal states, a set 'I' of input and a mapping 'f' of the product I x S into S (organisation is defined by specifying states and conditions). Self-organising systems (those progressing from lower to higher states of complexity, as in many social organisations) are not well suited to this approach, as their change comes from an outside agent. Our anti-virus strategy system is such a system and for this reason we will use the empirico-intuitive methodology.

Classical system theory uses classical mathematics to define principles which apply to systems in general or to subclasses. General System theory can be called the doctrine of principles applying to defined classes of systems. It is our hope that we can stimulate thought on how already-known principles can help us in managing our anti-virus protection by examining the system as a whole.

Return to Top

Holism

Our definition of holism, drawing where appropriate from the medical profession, is health-oriented, and focuses on maintaining and improving the existing health of the system. It does not focus on disease and illness. It is interesting to note that, while we have many terms that relate to compromised and infected systems, we do not seem to have many terms relating to

'well' computers. Holism operates under the assumption that the open system possesses an innate organising principle, with the interdependence of the parts having an effect on the total system health. Holism views symptoms of distress as signalling disharmonic conditions, from which we can learn how to adjust the system (feedback); it is open to a variety of approaches for attaining balance. The focus of holism is heavily slanted toward the correction of causal factors, not symptomatic relief. Thus, the role of the holistic practitioner is to facilitate the potential for healing [7].

Return to Top

Anti-Virus Strategy Systems

Where do our anti-virus strategy systems fit in this picture? We hope to explore some answers to that question by first examining the components of our model system. Keep in mind, however, that the goal of this paper is not to provide you with answers, but rather to stimulate new ways of thinking about the problems we face daily.

Return to Top

Components

Each of the components in Diagram 1 contributes to the overall health of the system. Conversely, each can contribute to the illness of the system. For instance, our computer can contribute to the health of the system by functioning properly. If the hard drive crashes, a disharmonic condition is introduced. Our managers contribute to the overall well-being of the system, as long as they perform correctly. However, if one of them intentionally or unintentionally infects a computer with a virus, he or she contributes to the illness of the system. Our software contributes to the wellness by keeping employees reassured, and by keeping viruses out. If it is disabled by an employee desirous of more speed upon boot, or if it does not do its job in virus detection, it contributes to the illness or chaos in the system. There are other factors not shown, as the anti-virus strategy system model does not stop at the boundary of the company. The model includes your Internet service provider, virus writers, makers of electronic mail front-ends, anti-virus product tech support people and more. For the purposes of this paper, we must draw an artificial boundary. We mention the rest to give you food for thought, and to illustrate that boundaries are not static.

Figure 1. Anti-virus Strategy System - The Environment

Return to Top

Programs Policy and Procedures

(Selection, Implementation and Maintenance)

Where do we begin in examining the interaction of our chosen system elements? Let's start with the software selection. Anti-virus software is selected based on a wide number of criteria (8). While some of these criteria are beneficial, several are counterproductive at best (9). We need to be aware of exactly how our company's software is being chosen, and not leave this vital aspect of software selection up to people who do not have the experience or expertise to make a selection that will maximize your organisation's protection against viruses.

Does your anti-virus software detect all of the viruses which are a real threat to your organisation? Before you glibly answer yes, you should recognise that all products are far from created equal, and that even the best products will not achieve this goal if not properly maintained. Consider the following:

> When asked what happens to two blocks of copper initially at different temperatures left alone together in an insulated container, students will reply that the blocks will come to the same temperature. Of course, if asked how they know, they usually say "Because it is a law of nature"...the opposite is true...it is a law of nature because it happens.[10]

Apply this to your anti-virus software. Does it catch viruses because it is anti-virus software? If so, you can depend on it, as its name defines what it is. But, if you even loosely apply this concept, you will see that it is anti-virus software because it catches viruses - and if it does not, then what does that make it?

Remember the following quote:

> 'If you call a tail a leg, how many legs has a dog?'
> 'Five?'
> 'No, Four. Calling a tail a leg doesn't make it a leg' [11]

Maintenance of your software is another critical issue. Maintenance refers not to the upgrade, but to the maintaining of the software on a daily basis. What does it require to run? Are you supplying what it needs to live? Or is it merely surviving? Does it have adequate memory, power, disk space to run optimally and lessen the chance your employees will disable it? Is it in an environment free from other programs which may hinder its performance? If you cannot answer yes to these questions, you are not providing an environment for this element of your strategy system which will allow it to remain viable. It will not survive. Like living systems, the anti-virus strategy system requires a favorable environment, else the system will adapt. Unfortunately, in the case of this system, adaptation can mean software becoming disabled by the user component of the system, or overridden by a competing software component. All this, and we have not even added viruses which by design cause a problem to the system by the introduction of instability.

Even if you have the best anti-virus software, and are running it optimally, there can still be problems. Software is just one part of the strategy system. Policies and procedures play an important role in the overall strategy. Even the viruses we mentioned earlier play a part in this system. Then there are the least predictable aspects of the system, the human beings. How complex is this system? How much should we expect the people involved to understand?

Ackoff defines an abstract system as one in which all of the elements are concepts, whereas a concrete system is one in which at least two of the elements are objects [12]. As you can see, our system is concrete. It is also by design an open system, one into which new components may be introduced. Some of these components are by nature 'unknown' (i.e. actions of people, how software may react, viruses which may appear).

When these components are introduced, we have to consider first how they behave on their own. Next, we have to consider how they would behave in combination with any and/or all of the other elements. Finally, we have to consider how 'things' in general will be if neither of

the objects are present. In its most simple form, a two-part system would require four equations, but of course, you can see that as the number of elements increases, the number of interactive equations grows by leaps and bounds [Table I].

	Linear Equations		**Nonlinear Equations**			
Equation	One Equation	Several Equations	Many Equations	One Equation	Several Equations	Many Equations
Algebraic	Trivial	Easy	Essentially Impossible	Very Difficult	Very Difficult	Impossible
Ordinary differential	Easy	Difficult	Essentially Impossible	Very Difficult	Impossible	Impossible
Partial Differential	Difficult	Essentially Impossible	Impossible	Impossible	Impossible	Impossible

Table I. [From [5]] - Introduction of Elements

One of the systems theory approaches we can draw from here to help illustrate the problem comes from what is sometimes called the Square Law of Computation. This means basically that unless you can introduce some simplifications, the amount of computation involved in figuring something out will increase at least as fast as the square of the number of equations. Consider all of the interactions between humans, computers, and software, and you will see why it is impossible to precisely calculate what the results of all of those interactions will be. We cannot even measure them. In other words, you cannot possibly anticipate all of the problems you will encounter in trying to keep your company's data safe from viruses, because you cannot possibly calculate the interactions which will occur once you begin trying to formulate a strategy. Needless to say, these interactions create 'problems'.

If we examine our anti-virus strategy in various ways, we may be able to see things more clearly. Another helpful way in which we can view our system is as an expression, such as the terms of a set. For instance, the notation:

Let x stand for marriage Let y stand for carriage Let z stand for bicycle

The set [x,y,z] is simple enough for anyone to understand. Using names in sets takes us to the more complex:[The look on your face when you saw your first child, a proof that Vesselin Bontchev is not the Dark Avenger, an atom of plutonium]; wherein the first no longer exists (or possibly never did); the second has not yet existed, and the third is out of reach of the common man.

If you were to be asked for the meaning of the ... in the set [Alan, Dmitry, Fridrik...] would you say the ... represented men's names? Names of programmers? Names of programmers who make anti-virus software? Names of people not from the United States?What is the rule for determining the meaning of what is unstated? Is there some unwritten heuristic of which your employees are not aware? What is the meaning of the three dots in our set?

This has a particular application to policy. Users can easily understand, 'Do not turn the computer off if you find a virus'. Can they as easily understand, 'Do not reset the computer if you find a virus'? Can they understand, 'In the event of a suspected virus, call the

administrator or take appropriate action'? What is a suspected virus? Is it any time the computer system seems to act strangely? Is it only when the letters fall off? After all, that's what viruses do, right? What is appropriate action? [Turn off the computer, Call your supervisor, Reboot the computer, ...] What is the meaning of the ... in this set?

Return to Top

Variations on a theme

How well are our strategies doing? As pointed out early on, not very well. Why not? To help answer that question, next we will examine the problems of our strategy using the concept of variation. We recognise the duality of variables as they relate to information processing; the significant values which variables acquire at the two extremes of their respective spectra. Specifically, in order for a system to continue to thrive, information must be processed. Disorder, uncertainty, variety - all must shift from high to low [Table 2].

Disorder, Uncertainty and Variety: Entropy and the Amount of Information Processed

High	**Disorder**	Low
High	**Uncertainty**	Low
High	**Variety**	Low
Large	**Number of Alternatives**	Small
Small	**Probability of an Event**	Large
Low	**Regulation and Control**	High

Table 2 - Predictable Output

The probability of particular events follows by decreasing from small to large. The amount of regulation and control increases from low to high. We become increasingly sure of the output of our systems [13]. However, viruses introduce a form of disorder with which the human components of our systems are not intimately familiar. While the probability of infection can be calculated mathematically [14], we are unable to calculate the probability of other events related to viral infections[15]. In what ways does this introduced unfamiliarity manifest itself? One manifestation is the appearance of problems.

We typically try to solve most of these problems deductively, to determine the reason for a variation between design and operation or design and implementation. This approach is doomed to failure because it places the blame on the subsystems. We attempt to 'restore to normal' instead of redesigning our system. We formulate plans based on incorrect, incomplete or obsolete assumptions. We neglect to factor in spillover effect, that is, the unwanted effect which actions in one system can have in another. Improving an isolated system may seem the epitome of system integrity. You can have your pure clean computer. Of course, it is virtually useless, unconnected to the rest of the world. Or, perhaps it is the solution. Isolated perfect machines. This would probably create a dissatisfied workforce, however, which would ultimately impact business negatively. In the case of anti-virus strategy, 'spillover' takes on many new dimensions - as many as the human beings with which our machines interface. Can you control all of the aspects of this system? You cannot.

Another factor to consider is the size and extent of our system. Further insight may be gained by considering what is sometimes referred to as the generalised thermodynamic law, which states that the probable state is more likely to be observed than the less probable. While this may incite the physicists among us, it has two parts which correspond to the first and second law of thermodynamics. The first law is hardly worth mentioning (physical reason), but the second is of interest to us. We should be concerned with the limited power of observers when viewing large systems. In other words, we cannot expect our managers to be in every place at once, knowing what is going on with every system, every employee. The concept of boundaries can be used to help solve this problem, but their definition is beyond the scope of this paper [16].

Return to Top

System Failure and Measurement

We say the system is failing for three reasons. It is not performing as intended. It is producing results other than expected. It is not meeting its goal. The objective is **NO VIRUSES**. However, in addition to often neglecting to define what 'no viruses' actually means, we are frequently unaware of how 'no viruses' can mean different things to different people. Not performing as intended could mean it finds some viruses but not all, or it finds all but only removes some. Unexpected results could mean it crashes 1 out of every 6000 machines, or produces system degradation you did not anticipate (if this is the case, does the fault really lie with the product for producing the degradation or you for not anticipating?) Not meeting its goal most likely means failing to keep out viruses. However, to some people, this is a different goal from 'no viruses'.

How is this possible? Isn't 'no viruses' a simple concept? In a word, no. When there is a malfunction, i.e. a virus is found, the natural tendency is to look for the cause within the system. We tend to blame the problem on the variation of the system from its 'desired' behaviour. It could be the fault of the program, the employee, the policy. We tend to blame the program as it is the part of the system most closely identified with the failure as immediately perceived. However, consider for a moment that, to your employee, 'no viruses' means simply that. No viruses are found. Following that line of thought, finding 'no viruses' would be a system success - that is, until it brought your operation to a halt. You see, to some people, 'no viruses' means that none are seen or observed, and not that none are actually operational in the system. We plan grandiose policies and procedures around finding a virus and make no space for 'no viruses' as a possible failed variation. If you find 'no virus', you need to be very sure it is not due to your employees disabling your software, or your software not finding the virus.

Many system 'improvements' are possible which in reality doom the system. Faulty assumptions and goals are often at the root of this problem. For instance, it is obvious that all of your computer workers must, under dire penalty, refrain from bringing disks from home into your office. You implement this policy. You assume they will comply. Your goal is compliance, not 'no viruses'. If the goal was 'no viruses', you would be forced to be more realistic.Consider the following two statements:

> We have clean, working computers and by not bringing in software, we can keep them that way. It will save us all a lot of time, and effort!
>
> If you bring in disks, you will probably infect our office computers. It will cost

us all a lot of money.

In the first instance, the focus is on the well machine. Everyone wants well machines. People like to be part of winning teams, and participate in things that are nice.

In the second, the focus is on the sick machine. None of your people would have viruses on their home computers. So, this must not apply to them. And if they do break the rule, you have already set them up to be afraid to tell you. After all, they don't want to cost you a lot of money and they certainly don't want to be known as the culprit for infecting the office computers.

How do we measure the performance of our anti-virus strategy system? Not very well. If we find some viruses, we say it's working. If we don't find any viruses, we say it's working. In some cases, you can apply 'we say it's not working' to these same sentences. There is no standard way in which we measure the success of the entire system. Only in the act of being out of control will the system be able to detect and bring back the control.

Return to Top

Conclusion

The systems approach proposed here is a 'whole system' optimization. Think of it as the configuration of a system which will facilitate optimal performance. There exists, of course, a dilemma, in that at some time suboptimization may be necessary, or even the only possible approach. An approximation which is used may be a great deal better than an exact solution which is not [17]. Nevertheless, our model will attempt to show ways to optimize system performance. Models are how we express things we want to understand and possibly change, designed in terms of something we think we already understand. Models sometimes present problems when you try to translate them into real world activities. With this in mind, I would like to suggest a simple model which may help us begin to find ways to find a solution to the problem of designing a workable anti-virus strategy.

> 'Models should not so much explain and predict as to polarize thinking and pose sharp questions.' [18]

Using a holistically modelled approach, we would strive to maintain the existing health of the system. This assumes we have a healthy system to begin with. This requires you not depend on your belief that your software is correctly installed and operational, and that your employees know how to use it and are using it, and that your equipment is functional, and that your policies are correct and being followed... It requires that you actually take it upon yourselves to designate people to ensure that your system is optimal to begin with. If you are not willing to do this, you cannot expect to restore the system to health. The focus should shift from 'blame' to 'responsibility'. This may require investment on your part. You may need to update equipment. You may need to train employees. You may need to purchase software. You may need to subscribe to publications which can keep your employees up to date on trends in virus and security matters.

You will need to monitor feedback between various aspects of your anti-virus strategy system. We have not discussed feedback at any great length in this paper, due to the number of elements of the system and the complexity of the feedback. However, using the empirico-intuitive General Systems theoretical approach defined earlier in this paper, you should be

able to determine the sorts of feedback which are required to keep your system functioning optimally. If there is NO feedback, you can rest assured your system will fail. Lack of feedback produces entropy. In simple terms, entropy can be called the steady degradation or disorganization of a society or a system. This is not what you want for your system. You want to move the system into organisation and order, high rates of probability and certainty. As we discussed earlier, this happens when information is processed. The information can be communication of any type between any elements of the system.

Our current focus seems to be on the existing illnesses in our systems. If open systems indeed, as suggested, possess an innate organising principle, perhaps we should be paying more attention to what the elements of our systems are telling us. We could learn the sorts of information required to maintain organised reliability. We could learn the amount and types of feedback required to process information optimally, and to keep the system both desirably adaptive and from adapting negatively. We must examine our systems as a whole, including all of the parts, as best we can, to determine what the elements and the system are telling us. In the case of our anti-virus strategy systems, we have yet to determine what that message is. Many of us have not even yet defined the elements of the system, the system boundaries, or the goal of the system.

It is clear that there are disharmonic conditions in the 'Anti-virus strategy systems' of most companies; if there were not, no one would be attending this conference or reading this paper. It is also clear that the way we traditionally approach these problems is not working. We have been using these approaches for a long time, and the problems are not going away. Drawing from the holism model, one thing we can do is examine causal factors, instead of focusing on symptomatic relief. We need to examine more closely the interdependence of the parts of our system, and as security professionals, should facilitate the potential for healing our systems. It is hoped that some of the ideas mentioned in this paper can provide a starting point for this.

The author would like to thank Louise Yngstrom, University of Stockholm, for late night chats on System Theory, above and beyond the call of even academic duty.

Return to Top

Bibliography

1. 'Virus Encounters, 1995: Cost to the World Population'. Testimony, House Subcommittee on Telecommunications and Finance, Tippett, Peter, June 1993.
2. 'The Wildlist'. Maintained by Joe Wells.
3. 'Real World Anti-Virus Product Reviews and Evaluation'. Gordon, Sarah and Ford, Richard, Proceedings of Security on the I-Way, NCSA, 1995.
4. 'An Introduction to General Systems Thinking', p.3, Weinberg, Gerald. John Wiley and Sons, 1975.
5. 'An Introduction to General Systems Thinking', p.51, Weinberg, Gerald. John Wiley and Sons, 1975.
6. 'General Systems Theory: Foundations, Development, Applications', pp.xix-xx, Revised Edition, von Bertalanffy, Ludwig. George Braziller, Inc, 1980.
7. 'Health Promotion Throughout the Lifespan', Edelman, Carole and Mandle, Carole. Mosby, 1994.
8. 'Guide to the Selection of Anti-Virus Tools and Techniques'. Polk, T. and Bassham, L. NIST Special Publication 800-5. NIST, December, 1992.

9. 'Real World Anti-Virus Product Reviews and Evaluation', Gordon, Sarah and Ford, Richard. Proceedings of Security on the I-Way. NCSA, 1995.
10. 'Semantics, Operationalism and the Molecular-Statistical Model in Thermodynamics', Dixon, John and Emery, Alden. American Scientist, 53, 1965.
11. Quote attributed to Abraham Lincoln.
12. 'Applied General Systems Theory', p.39, Van Gigch. John P. Harper and Row, 1974.
13. 'Applied General Systems Theory', Figure 2.2, Van Gigch. John P. Harper and Row, 1974.
14. 'Directed Graph Epidemiological Models of Computer Viruses', Kephart, Jeffrey O. and White, Steve, R., Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, 1991.
15. 'The Viability and Cost Effectiveness of an 'In the Wild' virus scanner in a Corporate Environment', Gordon, Sarah, 1995.
16. 'Applied General Systems Theory', p.25, Van Gigch. John P. Harper and Row, 1974.
17. 'The Development of Operations Research as a Science', pp.59-60, as cited in [4]. Ackoff, Russell. Scientific Decision Making in Business.
18. 'Some Mathematical Models in Science', Kac, Mark. Science, 166 No. 3906 695, 1969.

Return to Top

About the Author

Sarah Gordon's work in various areas of IT Security can be found profiled in various publications including the New York Times, Computer Security Journal and Virus Bulletin. She is a frequent speaker at such diverse conferences as those sponsored by NSA/NIST/NCSC and DEFCON. Recently appointed to theWildlist Board of Directors, she is actively involved in the development of anti-virus software test criteria and methods. She may be reached assgordon@low-level.format.com

back to index

WHAT IS WILD?

Sarah Gordon
IBM TJ Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598
sgordon@watson.ibm.com

Abstract

"In the Wild" virus detection is part of the criteria of *National Computer Security Association (NCSA)* Anti-virus Product Certification, *SECURE COMPUTING Checkmark* Certification, the proposed *UK IT Security Evaluation and Certification (ITSEC)* anti-virus product certification and other product review and evaluation schemes. However, companies which use "certified" products, based on "In the Wild" (ITW) detection continue to suffer the effects of viruses. This paper considers the various definitions of "In the Wild", as well as how well the "In the Wild" criteria as defined by the individual testing organizations measure the ability of products to deliver adequate protection. Inherent problems with such approaches are discussed from both a development and user perspective. Some alternative testing, development and protection strategies are offered.

Introduction

There are currently over 10,000 computer viruses in existence. Most of these have little likelihood of spreading and exist only in collections; they are known as "Zoo" viruses. Even an anti-virus virus researcher would be hard pressed to list a significant percentage of these viruses, let alone provide detailed information on how they operate. Most users have only even *heard* of a handful of them. Yet when a virus is encountered within a company, it is usually the case that a call to an anti-virus vendor, or a search through a virus encyclopedia will provide further information on that particular virus. This is because vendors, researchers and testers have begun to focus their attention on those viruses "In the Wild".

The concept of "In the Wild" is an important one. Tests of anti-virus software have, until recently, focused on Zoo detection figures. These tests did not necessarily measure the ability of a product to meet the real world threat [1]. Consider two products tested against a corpus of infected files: by simply measuring which product detects more infected samples, we would be given no information concerning how well the product detects and repairs those viruses which are known to pose an active threat to a real world PC. A meaningful test of the efficacy of a product would be to measure the product's ability to detect and remove *those viruses that the user is likely to encounter:* that is, *those viruses that are "In the Wild"*.

In order to understand the issues surrounding "In the Wild", we will examine a history of the term. As far as we can determine, the actual phrase "in the wild" was first used informally to describe real-world virus incidents by Dave Chess, of *IBM's TJ Watson Research* facility, in 1990/91 [2]. Around this time, Alan Solomon remembers using the term in telephone conversations in the UK [3]. The phrase subsequently cropped up in *Virus Bulletin* in 1992, in a message from Roger Riordan [4], where he referred to real-world incidents: "As Dave Chess pointed out on Virus-L (May 8th, 1991), few of your specimens have ever been seen in the wild..." [5]. It formally appeared in 1992, in "Measuring Computer Virus Prevalence"[6] where it was shown that a small number of viruses accounted for most actual virus incidents, i.e. were "in the wild". Early *Virus Bulletin* tests featured an "In the Wild test-set", a collection of viruses designed to measure real-world performance. The contents of this list were garnered from virus reports sent to *Virus Bulletin*, along with those viruses which researchers believed to be spreading, as opposed to those which were known to exist but which were not observed to be spreading (Zoo samples). While this was not entirely scientific, it is the first test of which we are aware that made a reasoned and logical attempt to move away from Zoo testing. During this time period, most new viruses were not initially discovered spreading in the wild and the vast majority of Zoo viruses were not considered to be an active threat. At the same time, Zoo testing remained prevalent [7] and users tended to judge products based on how many viruses the product could detect.

There were some obvious problems with this approach. As the number of viruses "In the Wild" continued to rise, the need for a better definition of "In the Wild" led to the creation of *The WildList* by Joe Wells [8]. Wells' idea

was to take the concept of "In the Wild" used by *Virus Bulletin* and expand it internationally. To this end, he culled virus reports from virus researchers worldwide. Any virus that was reported by two or more researchers was considered to be spreading in the wild. In some cases, viruses were reported by only one contributor. Those viruses were placed into their own section of *The WildList*. (This supplemental list provides some idea of which viruses might be moving onto or off of the main section of *The WildList* and tends to be a more regional reporting mechanism.) A supplemental frequency table has been recently added. It does not show how common each virus is. Rather, it is *The WildList* sorted by the number of participants that report each virus. It gives the names, types, and aliases of the most frequently reported viruses. These viruses have been reported by at least one third of *The WildList* participants. They are sorted with the most frequently reported first. *The WildList* clearly states it should not be considered a list of the most common viruses, as no commonness factor is provided; however, it can serve to help determine which viruses are spreading.

The WildList represents the most organized effort to catalogue those viruses which are spreading, yet it would be wrong to define "In the Wild" as those viruses which are listed in *The WildList*. Rather, *The WildList* should be considered to be a subset of this set - a core set of viruses that every product *must* be able to detect. Making this subset more complete is fraught with problems, and falls prey to different definitions of what "In the Wild" actually means. Before considering the impact of tests based on *The WildList*, we shall examine some of these problems.

- ***The WildList* lags behind viruses spreading in the wild.**

Time delay is by far the most commonly reported complaint against using *The WildList* as a complete set of viruses "In the Wild". Put simply, the logistics of compiling reports from more that 45 contributors worldwide, who in turn have to compile their own list of Wild viruses based upon their technical support calls, can quickly date the contents of *The WildList*. In practical terms, a virus may make it onto *The WildList* two or even three months after it is first discovered at a user's site. At present, this is difficult to improve upon, though streamlining analysis of submissions may help. Procedures to facilitate this have now been implemented, by way of automatically distributed, standardized reporting forms for participants, which will decrease the time required to process incoming submissions. These new forms should make reporting much simpler for the volunteer reporters, and it is hoped this may help aid in getting submissions in faster.

- **The viruses on *The WildList* are those viruses reported, not necessarily those viruses which are in the wild.**

As *The WildList* contributors are mostly made of those working within the anti-virus industry, it is not unreasonable to assume that *The WildList* represents those virus infections which are reported directly to developers/resellers. However, this group of viruses is not necessarily a complete list of viruses "In the Wild". Consider the case of a hypothetical virus named Foo, spreading in the wild. This virus is detected and removed perfectly by all anti-virus products. Also, consider another virus spreading, called Bar. When a certain product detects Bar and attempts to remove it, it corrupts the file. It is entirely believable that anti-virus companies will received far more reports of the Bar virus than the Foo virus, even if they have equal prevalence. Thus, researchers may preferentially receive reports of those viruses which are not adequately dealt with by anti-virus products.

One apparently obvious solution to this bias this would be to include businesses (as opposed to solely developers, resellers and researchers) in *WildList* reporting. However, although many companies require that all virus incidents are reported to a central body, some studies of computer virus epidemiology strongly suggest there are other problems with organizational reporting. Indeed, it is possible that problems with corporate statistics can reflect the inclusion of wild guesses, reporters being unaware of virus incidents and reporting bias toward problematic viruses [9].

- **The samples named in *The WildList* may not be the same viruses actually spreading in the wild**

This question highlights one of the areas which is actively being improved within *The WildList*; correlating reports of viruses *by name* with actually binary samples of infected files. It is believable that discrepancies over virus naming could both lead to viruses being inadvertently added to *The WildList*, as well as viruses being omitted.

Consider the case of a single virus, Foo, which is identified by product A as Foo.A, and identified by Product B as Foo.B. In such a case, if researchers simply correlated reports from the field with their own virus collections, both Foo.A and .B might be placed on *The WildList*. Next, consider two viruses, Foo and Bar, which are both identified by products A and B as Foobar. In such a circumstance, only one virus name would be added to the list of those viruses known to be "In the Wild", where in actuality, there should be two different entries.

To help solve this problem, Wells has added additional criteria to those contributing to *The WildList*: if a sample is being reported for the first time, the reporter must supply a sample of that virus along with his report. This allows submissions of the same virus reported by different name to be caught, and similarly allows one to discriminate between two different viruses inadvertently identified by the same name. *The WildList Organization* has begun distributing the responsibilities associated with *WildList* sample replication and identification of the viruses amongst *The WildList* board members, and is including a peer review process for samples. It is hoped this will significantly decrease the workload and increase the naming accuracy.

***Note that even though we have outlined a number of shortfalls in *The WildList*, the author still believes that it is currently by far the best resource for tracking those viruses which are believed to be in the wild.**

Tests Based Upon *The WildList*

Using confirmed samples of every virus on the list as a test suite for testing anti-virus software can tell you whether or not a product detects the viruses on the list. While this is clearly only a minimal test, by monitoring the tests over time we should theoretically be able to determine whether or not a vendor continually meets the test conditions. The problems with practical aspects of doing this will be addressed later in this paper. Also, it is important to remember that while *The WildList* clearly defines what is meant by "In the Wild" for the purposes of the list and for tests which use the list, it is not a definitive measure of viruses that are causing incidents. For example, certain viruses have been found spreading in isolated areas of the world. While they are definitely in the wild causing incidents, the fact that they are reported by only one person keeps them from being included in the main section of *The WildList*. This creates a problem for users who rely on ITW-based certifications as the only measure of a product's effectiveness, because current certification schemes, as will be shown later, do not test against even the upper portion (viruses in the wild, reported by 2 or more contributors) of the most current release of *The WildList*.

In this section, we will briefly examine three testing/certification schemes, before discussing how much assurance each gives to the user that his/her product will truly detect those viruses which are known to be spreading. Interestingly, each scheme uses a slightly different definition of "In the Wild" for the purposes of its tests. With this in mind, we will then re-examine *The WildList* as the baseline measure used by several certification bodies.

NCSA Criteria

Founded in 1989, the *NCSA* is a for-profit organization based in Carlisle, Pennsylvania. Its anti-virus product certification began in 1992. The main thrust of the criteria currently is to provide a way to measure the effectiveness of detection capabilities of virus scanners. The scheme requires that the scanner components of certified products detect 100 per cent of viruses found on the upper portion of *The WildList*, using a *WildList* that is two months old at the time of testing. This is said to allow for development time. We will now briefly examine relevant aspects of the scheme.

According to documentation published on the *NCSA* World Wide Web site, "*NCSA* tests and certifies that anti-virus scanners pass a number of stringent tests." As our own tests have shown that some *NCSA* certified products should not pass the documented certification criteria, we asked *NCSA* for information regarding their virus test suite, to see if we could determine the cause of the discrepancy. At *NCSA*'s invitation we visited its' virus lab where several virus test suite related problems were noted. One problem we noted was related to the replication of polymorphic viruses. Some viruses attempt to hide from virus scanning programs by keeping most of their code garbled in some way, and changing the garbling each time they spread. When these viruses run, a small header "de-garbles" the body of the virus and then branches to it. A polymorphic virus' de-garbling header changes each time the virus spreads. The polymorphic test-set had not been fully replicated; only 6 viruses had been replicated. As some products may have unreliable polymorphic detection, a more complete polymorphic test suite is desirable. Ideally, all polymorphic viruses "In the Wild" should be replicated onto appropriate hosts, but this is a difficult and time-consuming task. Additionally, some viruses are multi-partite, which means they are capable, for example, of infecting not only files, but Master Boot Records of hard disks, or floppy disk boot sectors. These types of viruses should be replicated onto all appropriate media; we observed that this is not being done in *NCSA* tests at this time. Again, this replication presents some unique problems and will take some time to sort out. The macro virus test-set consisted of two replications of each macro virus except for ExcelMacro.Laroux, of which there was only one sample. This number of replicants is insufficient to allow for measurement of the reliable detection of macro viruses. There were not any macro viruses replicated onto *Office97* documents. As some of the macro viruses will replicate upwardly into *Office97* documents, inclusion of such documents is required in order to measure the level

of protection afforded the user. Nine of the boot sector viruses had not yet been replicated; work is currently in progress to rectify this. It can be difficult to replicate some of the viruses, requiring special expertise and in some cases, additional equipment. Plans are underway to initiate testing of master boot records of hard disks; however *NCSA* is some time away from this type of test. *NCSA* has demonstrated its commitment to fully expanding its test suite. *NCSA* virus lab technicians are currently in the process of solving these virus related problems by replicating the macro and polymorphic viruses into a larger suite, and by replicating the multipartite viruses onto appropriate media. According to *NCSA* spokesperson Jon Wheat, these issues are a "top priority".

Some administrative problems should be noted. Here, we are concerned with the timeliness of the tests and consistency of the scheme. On March 17th, 1997, the *NCSA* Web Site listed *F-PROT Professional 223a* as a certified product. Version 2.23a was released in August 1996. Similarly, *Eliashim's ViruSafe* version 7.1 was shown as certified; however, the most current version of *ViruSafe* on March 17th was 7.3. Two *Intel* products (*LanDesk for NT* and *LanDesk for NetWare*) were shown as last being tested in December 1996. *Dr. Solomon's* software versions 7.65 were the most current tested versions; their current release on March 17th was 7.69. When asked about the discrepancies, *NCSA* stated these products would be retested. As documented in [10], the scheme has not been without problems; however, at the time of the completion of this paper, these problems appear to be in the process of being resolved, and products appear to be tested on a regular basis.

NCSA is working hard to make the tests of the scanners complete, thorough and accurate and to keep the administrative aspects of the scheme functioning smoothly.

SECURE COMPUTING Checkmark

SECURE COMPUTING magazine, published by *Westcoast Publishing*, regularly reviews and evaluates anti-virus software, provides a venue for marketing of various security software products, and offers security-related articles for its subscribers. A recent addition to these services is the *SECURE COMPUTING Checkmark* scheme, which is designed to establish a standard for computer security products, test them against that standard and produce a certificate which shows that they meet the standard. They claim this is similar to governmental standards that seek to indicate to a buying public whether they can have faith in certain products. According to the *SECURE COMPUTING* Web page:

> "Whatever your needs, you should know that the products you are buying are worth the money and give you a sensible level of security....It shows that the product has been tested and approved to an industry-recognized standard by an independent organization....There are also one or two other schemes run by private companies or by students in universities but in our opinion these are not worth bothering with."..."To obtain a Checkmark an anti-virus product has to detect all the viruses that are in-the-wild; that is those which are actually out in the real world causing infections (not held in the private collections of anti-virus researchers). The Checkmark in-the-wild list is updated on a monthly basis. Products are tested on the basis of the in-the-wild list current three months previously and there are a number of practical reasons for doing this. It is a fairly typical approach and in the real world gives a high standard for the anti-virus developers to achieve."

As with the *NCSA* library, not all multi-partite samples have been replicated onto both files and boot sectors; this process is underway. Tests of boot sector infections are done on real infected floppies; there are as yet inadequate resources for testing of Master Boot Records of virus infected hard disks. There are a thousand replicants for each polymorphic virus which has been replicated so far. *Secure Computing* will be including *Office 97* capable *Word* viruses now that some have appeared on *The WildList*; they are not testing any of the upwardly mobile *Office 95* viruses on *Office 97 Word* goat documents until these viruses are explicitly reported on *The WildList*.

This scheme could attempt to address the problem of *The WildList* lagging behind the current threat as it retains the option to add viruses at the discretion of the test administrator [11]. It remains to be seen if this results in tests which provide a good measure the protection provided; however, all indications are that the tests should be thoroughly and competently performed. *Westcoast Publishing* is well positioned to promote the scheme in both the United States and Europe, using *SECURE COMPUTING* magazine as well as its' recently purchased *InfoSecurity News*.

***ITSEC* Certification**

The proposed *UK IT Security Evaluation and Certification (ITSEC)* model of anti-virus software certification consists of several criteria, all of which are designed to measure how the product meets the dynamic real world threat. Several anti-virus product vendors have been involved in helping draft guidelines, and the specialty

magazines *Virus Bulletin* and *SECURE COMPUTING* have also sent representatives to *the Anti-Virus Working Group* meetings. The evaluation process is in the developmental phase although significant progress has been made in the past year, particularly in the area of formalization of criteria. The main areas with which the process is concerned are Standard, Threat Assessment, Virus Attack Techniques, *Anti-Virus Working Group* Virus Collection, Comprehensive Virus Collection, "Advice Documentation", and Certificate Maintenance Scheme.

With the *ITSEC* scheme, an increasing level of stringency would be applied and associated with the commonality of the virus or observed technique, i.e. weighted testing. The current plan is to perform tests with common and wild viruses (note that *ITSEC's* definition of "In the Wild" is not the same as that used by *The WildList*) listed concurrently and cumulatively and to require a 100% score to pass. Common viruses are defined by the *ITSEC* scheme as those which are frequently reported as causing attacks; it defines "In the Wild" viruses are as having been recorded as responsible for attacks. Determination of which are common and which are "In the Wild" is to be made by a national authority that monitors the changing situation reported from worldwide centers of virus expertise. The current strategy for Zoo testing is detection of at least 90% of the different named viruses in an approved collection for a passing score. The *Anti-Virus Working Group* recently announced that the University of Hamburg has agreed to act cooperatively with the *ITSEC* evaluation scheme and do Zoo testing on-site at the University, as part of the evaluation process. Other collections may be used, providing they meet certain requirements, yet to be determined. In addition to this detection criterion, a number of other proposed criteria are put forth in the formal documentation provided by the *ITSEC Anti-Virus Working Group* drafts for Standard Functionality Class for Anti-Virus Products. These criteria include, but are not limited to, areas including recovery means, false positive levels, common compression, self-checking, logging and naming. Discussion of these is beyond the scope of this paper.

A Virus Attack Techniques Encyclopedia has been developed (under contract) by the *Anti-Virus Working Group*. This document is intended to detail all known techniques used by viruses, and currently includes the following: boot record infectors, parasitic viruses, multipartite viruses, companion viruses, stored code modification, environmental format considerations, stealth, execution infectors, system infectors, interception of system services, defense mechanisms, payloads, permanent configuration changes, hardware/software specific viruses and macro viruses. It is a dynamic document. The encyclopedia will be used to help in formulating ways to more fully analyze and test products; for security reasons, it is a limited distribution document.

By attempting to measure a product's performance against the threat by scanning a comprehensive large collection of all viruses, testing extensively against those viruses which are known to be "In the Wild" according to designated reporting authorities, and measuring product abilities against a range of different attack strategies, the *ITSEC* scheme is focusing on the current and *future* "In the Wild" threat. By evaluating the product's ability to defend against the different techniques used by viruses, they hope to provide a measure of a developer's ability to track a rapidly changing threat. The *CLEF* would maintain close contact with the developer of the product currently under evaluation, with developers being required to demonstrate that not only are they up to date with the current threat, but that they have in place sufficient procedures to monitor the threat as a function of time and update the software to meet this threat. This would be documented through the use of the Certificate Maintenance Scheme, which includes extensive paperwork on the part of the developer to document their resources and plans in various areas including intelligence activities related to monitoring the threat, threat analysis and countermeasures. This "vendor evaluation" is something that almost no other evaluations of anti-virus software includes, and is one of the biggest benefits of the proposed *ITSEC* approach. It is also one of the areas which appears to meet with the most resistance within the USA. One concern which has been cited is the sharing of information between *CLEFs*: "Even though the UK requires that all techniques and lessons learnt from evaluations be documented at the end of an evaluation and made available to the UK evaluation community, it is felt that *CLEFs* prepare this information from a position of non-disclosure of information which is of a proprietary interest to them. There is some concern UK based evaluations, by virtue of their commercial nature, do not encourage the sharing of evaluation techniques amongst the evaluation community" [12].

There is another potential problem with this type of approach. As documented in [1], this solution can lead to possible problems as new threat types may be as yet unanalyzed, and the virus itself is not in the wild. There is no guarantee as to the time sequence that a virus may be found to exist, be found in the wild, be obtained and analyzed by an evaluation or certification service, and its threat type documented. This is illustrated by the recent spate of macro viruses, where initially there was a noticeable lag between the knowledge of the threat type by anti-virus researchers, the discovery of the first in the wild *Word* macro virus[13] and the first in the wild *Excel* macro virus

[14], and the implementation of detection and prevention for these virus threat types on the part of some developers.

Finally, there are problems with issues of legal liability. Whereas German law demands someone be liable for failure in *ITSEC* certified products, the United States makes specific disclaimers assuming no responsibility. Drawing again from Borrett [12], we find "the political implications of legal liability for Europe and North America merits further investigation. In the interim, it may suffice to place an appropriate caveat alongside any US evaluated products which appear in UK Certified Product List publications."

Future Trends: New Paradigms and Epidemiological Shifts

While each of these schemes use *The WildList* as a basis for wild virus detection only one (*ITSEC*) represents more than a series of snapshots of particular product's detection. We see several dangers associated with the current situation. In order to better illustrate these dangers, let us first build a perfect set of review criteria. Note that here we shall attempt to address only those aspects required for virus detection; properties such as virus removal, product usability and technical support are beyond the scope of this paper.

The shift from Zoo to "In the Wild" testing marked the beginning of a move towards measuring the protection provided by a product. However, this shift is only the beginning of a true measure of protection provided. A cursory examination of *The WildList* shows us that a particular computer is more "at risk" of infection by certain viruses on the list than certain others [15]. For example, Ping_Pong.B and Wazzu are both on *The WildList*, yet few would argue that for the average computer, the probability of infection with Wazzu is considerably higher. However, in most tests carried out against the set of viruses catalogued in *The WildList*, each sample is equally weighted. Clearly, this is not a complete approach; in a "perfect" world, we would weight each virus by the actual probability one had of encountering it and it effecting one's work. Extrapolating onward, we would include all Zoo viruses in this weighting; for example, those viruses which are difficult to replicate would have a low rating (not in the wild, and not likely to spread even if released), whereas those viruses which have been actively circulated in newsgroups and which are highly viable in the wild would have a higher rating. However, even this approach is not complete. It is easy to argue that while the overall features of such a weighting scheme for viruses would vary relatively slowly as a function of time, its details may fluctuate rapidly. Consider, for example, a situation where a certain virus is distributed widely on a set of mass-produced CDs. The threat posed by this virus (that is, the probability that you will encounter it) has increased somewhat, even though it may have only actually infected one PC at this time. Another layer of complexity which we will not address here is that such a weighting scheme would vary depending on whom the review was being carried out for; *Word* macro viruses, for example, pose little threat to those who do not use *Microsoft Word*.

Initially, we believed that testing based on criteria that involved this type of weighting was impossible. We have since determined that the tests could be done using data gathered from *IBM* studies. However, problems with formalizing such a scheme remain. While using the data to formulate test criteria that could measure threats on a global scale is feasible, we believe certification using these methods is not practical at this time, due to the need for the certification body to independently gather the necessary comprehensive data. Other methods of measuring real world virus prevention provided by a scanner need to be compared to this model. Making such a comparison, we observe that for each of the certification bodies we have examined, all fall short in terms of the currency of the viruses used for testing. At one time, we believed that this was not a serious problem [16]; however, recent shifts in the way viruses appear in the wild are rapidly altering this perspective.

The most serious change in the ways viruses spread since perhaps the beginning of the virus problem is posed by macro viruses. These viruses attach themselves to data items that are frequently shared. Moreover, this sharing is often done via the LAN or Email, making such macro viruses highly virulent. Indeed, macro viruses have been so successful in the wild that the two most reported viruses to *Virus Bulletin* in January 1997 were both macro viruses: Concept and Npad. We have observed that once a virus begins to spread rapidly, it can reach epidemic proportions within an organization very quickly. It is the combination of large spread rate and lag in *WildList* testing times of *WildList*-based certification schemes which poses the biggest problem to those relying on *The WildList* for certifications. Since a virus must be reported by two or more *WildList* contributors, it is possible for a virus to be rampant within one organization and still be observed by only one *WildList* reporter. By the time a virus discovered in the wild is actually observed by two reporters and included in the certifying body's test-set, the virus may have already been spreading within any given organization for several months. A good illustration of this is the Concept virus. Discovered in July 1995, the virus first appeared on *The WildList* on Sept 10th, 1995. Thus, by the rules of a certification body using the criteria of detection of a collection based upon a two month old *WildList* compliancy, a

product which was certified would only be required to detect this virus by Nov 10th, by which time it was already spreading rapidly in the wild.

Another problem has developed which may impact the ability a certification body has to measure that vendor's ability to meet the threat posed by macro viruses. A macro virus which replicates under *Office 95's* version of *Word* may be automatically converted by *Word* to the new *Office 97 Word* format. This is referred to as 'up-conversion'. The problem is related to a controversy within the anti-virus community regarding this up-conversion of *Office 95 Word* macro viruses and testing of anti-virus products. Some anti-virus researchers have indicated they feel this up-conversion of in the wild *Office 95* macro viruses is the creation of "new" viruses, and as such, represents an unethical act for any anti-virus product tester. Others researchers maintain the opinion that *Office 95 Word* macro viruses which are in the wild and able to replicate into *Office 97* documents, (via *Word*), should be part of the in the wild test set in both their *Office 95* and *Office 97* form, as to do otherwise could expose users of certified products to unnecessary risk.

Is such testing required to make sure users are adequately protected [24], as part of an *ITW* based certification, or would this be an unethical act of irresponsible virus creation? No one can argue that the *Office 97* viruses are in many ways different from their *Office 95* origins. However, we question whether this difference in physical structure, form, and language supports the contention that these are in fact totally different viruses and that replicating the *Office 95* virus onto an *Office 97* document is unethical virus creation. It is the opinion of this author that such arguments are counterproductive and that certification bodies which perform ITW tests and certifications should simply replicate *Office 95* macro viruses onto *Office 97*documents, using due diligence in the care of such samples, as part of these ITW tests and certifications. As an industry, the anti-virus industry has long held the position that virus creation for any reason is unethical. This belief has been somewhat altered by the necessity to perform tests of viruses generated by virus creation 'kits', and the need to generate multiple polymorphic samples to allow for reliable detection and disinfection. The evolution of the virus threat may force us to re-examine our beliefs yet again.

Another serious problem for certification bodies brought about by macro viruses is the vast numbers of variants we are observing coupled with the concept of "In the Wild". Virus exchange sites appear to be less prominent for macro viruses, than is the case for file and boot sector infecting viruses. The majority of these macro virus variants are being discovered already spreading in the wild. We believe that there are a number of reasons for this. First, as current macro viruses are written in *WordBasic*, they essentially carry around with them a complete copy of their source code [17]. As the language is both simple to use and powerful, viruses are easily modified and released. Second, we have observed seemingly random corruption of macros within the *Word* environment. While we are as yet unable to reliably recreate such corruption in a laboratory environment, we can see that macro viruses seem to be more resilient to such corruption than binary viruses. Thus, whereas a corrupted binary virus frequently renders a virus non-functional, many *Word* macro viruses are quite capable of replication even when corrupted, leading to creation of a new variant. Thus, we have observed certain *Word* viruses spawn many variants in just a few months - something which rivals even the most prodigious of "ordinary" viruses. This rapid rise of new strains discovered in the wild has further clouded the concept of "In the Wild," as well as reduced the value of certifications carried out against *The WildList*. A more forward-looking approach would appear to be that described earlier as taken by *ITSEC*, which attempts to certify a company's ability to meet the current and future threat. It would appear that in terms of protecting the user, the most critical question is no longer whether a company can detect a specific virus, but how quickly that company can meet a new threat.

Some people have argued that all viruses are effectively "In the Wild", as many virus collections are available via virus exchange bulletin boards and web sites. However, a virus which is found on a Bulletin Board System or web site may not be viable in the real world. In 1992-1993, we examined the relationship between viruses found on virus exchange BBS compared with those known to be causing incidents [18]. It was determined there was little if any reason to believe viruses on underground BBS contributed significantly to the population of viruses spreading in the real world. The majority of these viruses simply were not found to be spreading. At the same time, individuals were reporting (and continue to report) infections caused by some of these barely viable viruses; this may be a result of the users obtaining the viruses and using them for testing (or reporting) purposes.

In 1994, we began to observe a change in the nature of virus exchange and distribution. It was concluded that with the growth of the Internet, viruses could reasonably be expected to spread using several different models. Specifically, Web virus distribution was predicted to make viruses widely available to the general computing population should they desire to obtain them; Usenet news was shown to be a potential distribution media for

viruses for both the willing and unwilling, and the Internet itself was examined as a potential hotbed for viral spreading which could occur almost instantly and worldwide. "The system is the perfect medium to host and transfer the very programs designed to destroy the functionality of the system itself"[19]. Whether or not the increased availability and relative anonymity afforded by the Internet will contribute to in the wild virus population remains to be seen. Viruses that have been released via Usenet have not become rampant in the wild. However, certification bodies who rely on detection of those viruses "In the Wild" should keep a careful eye on the role of global inter-networking, lest they be taken unaware by a paradigm shift in the way viruses spread. We are already beginning to observe real problems in this area, which we will discuss later in this paper.

Threat and counter-threat

The need for a new method of reviewing and certifying anti-virus software becomes more apparent when we examine some of the new threats resulting from the increased use of networks and desktop Internet connectivity. Although we have yet to see a virus spread in minutes/hours on a global scale via Email, we believe that the potential for such a virus exists. There have been several precursors to such a virus; here we shall discuss two of them: CHRISTMA EXEC [20] and ShareFun [21].

CHRISTMA EXEC is a well-known "chain letter", which was released on December 9th, 1987. It is a good example of how an e-mail worm can impact a network: CHRISTMA EXEC spread across BITNET, EARN and *IBM*'s internal network, dramatically slowing the *IBM* worldwide network on December 11th, 1987. The program, written in REXX, spread on VM/CMS installations, and displayed a Christmas tree along with a message, before sending a copy of itself to all of the users' correspondents in the user files NAMES and NETLOG.

ShareFun.A is a macro virus which spreads by infecting *Word* documents, and as such, operates just like most other macro viruses. However, ShareFun.A attempts to spread via desktop e-mail, attempting to send mail messages to addresses listed in the users' address book. The message has the subject line "You have GOT to read this!", and it carries with it an attachment which contains the infected document. Fortunately, the virus e-mail routine is not very effective relying on certain applications being active upon the user's desktop, and so is not likely to be spread rapidly via this mechanism.

A virus, by definition, replicates, and attaches itself to a host program. Although CHRISTMA EXEC did not attach itself to a host and therefore was not strictly speaking a virus, and ShareFun.A appears to be flawed in its design, these examples of malware provide a definite warning of things to come. Collecting virus samples, extracting signatures and distributing cures have traditionally been time-consuming tasks for the anti-virus researcher. The upgrade and updating processes have required frequent action on the part of users. As we have observed more and more viruses, some anti-virus vendors have developed automated methods to deal with the analysis of common viruses. This has helped cut the workload, but is still insufficient to deal with the virus problems of the future. In a time when viruses can spread worldwide in hours or even minutes, a day or two of waiting could render a company impotent. Even automation of the distribution of signature updates via techniques such as push-technology will not fully solve the response-time problem; for viruses which spread chiefly by computer-computer interaction, rather than human-computer interaction, the interactive and time-consuming element of isolation, capture, replication, and analysis is quite simply too slow. We believe that current levels of protection are not sufficient to defend well against an e-mail-aware virus. By the time such a virus could be isolated, sent to researchers, replicated, analyzed, a fix provided and that fix disseminated worldwide, the virus may well have already reached epidemic proportions.

In an attempt to address this problem, *IBM Research* has developed a biologically inspired anti-virus technique: a computer immune system that can automatically identify, analyze and remove the virus from the system [22]. The immune system provides for automated collection and analysis of viruses, but does not stop there. It prepares and distributes the immunization for the virus automatically. No human intervention is required in most cases. Simply put, the immune system monitors activity and filters it for virus-like behaviour. If it is determined that a known virus is present, it deals with the virus appropriately. However, if a known virus is not found, the system then automatically transmits a copy of the suspected infection (via a transaction center) to the *IBM Research Division* labs. There, with no danger to the user's machine, the system releases decoy programs, which seduce the virus into attacking. The decoys are examined for modification, and when such modification is found, viral signatures are extracted, and a repair algorithm is generated. This algorithm is automatically distributed throughout the system, curing both the virus which has been found there and on any other machines which have enabled the immune system. At the same time, immunity to that virus is provided throughout the system. All of this can take place in a matter of minutes, making use of secured authenticated transactions between the users PC and the *IBM Research*

Division secure lab. Although human input may still be required in some rare situations, it is hoped that the ability of the immune system to respond to new threats will far exceed conventional techniques.

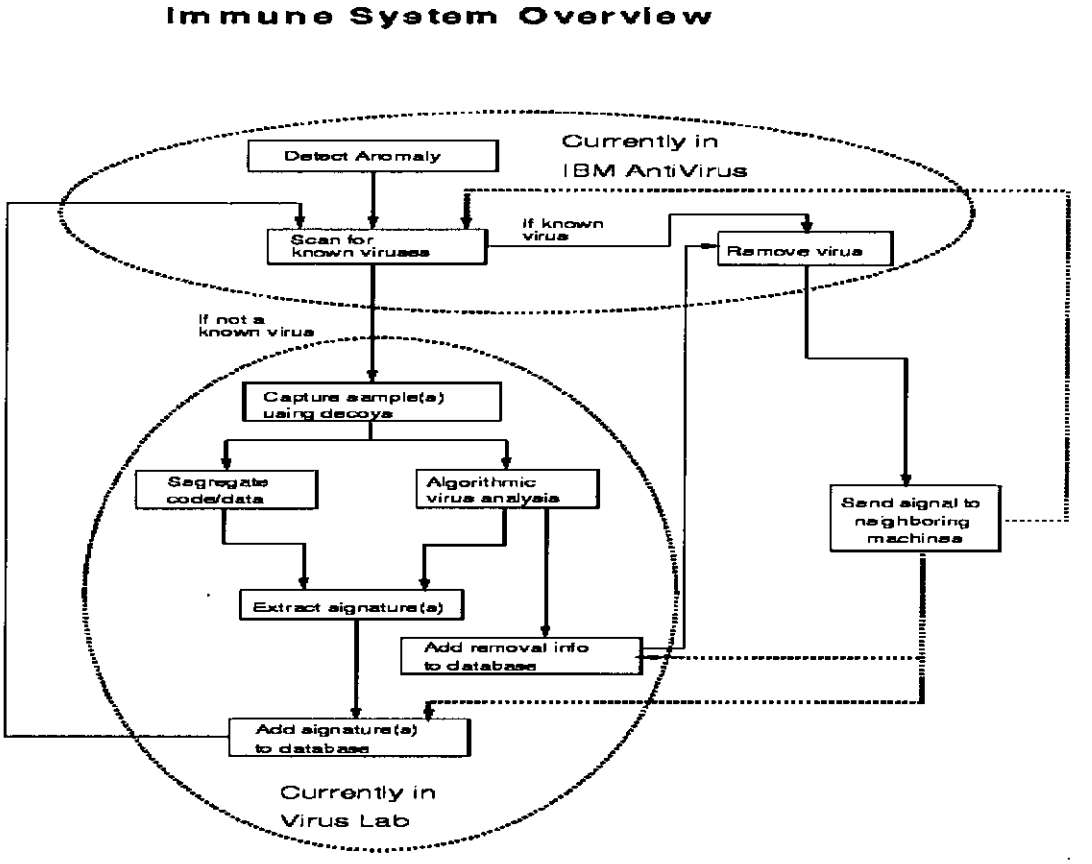


Figure 1. [26]

Certification Challenges

This immune system model offers obvious benefits to the user and administrator: detection, removal, product updating and product distribution to multiple sites are all done immediately and transparently. The benefits to the developer include freeing of time for the anti-virus researcher who no longer will spend time analyzing trivial viruses, and implementing their detection. However, this protection model offers both technical and administrative challenges to the certification body. The challenges are many; here, we will examine several of them.

Design Criteria and Test Administration

Currently, central to the test criteria for all anti-virus product certifications are various lists of viruses with obvious names. Indeed, as reflected in the *ITSEC* guidelines, naming compliance and consistency are sometimes important parts of the product [23]. The immune system model eliminates the need for developers to decide on names before detection and disinfection for new viruses can be implemented. *The WildList*, being name-based, would be unsuitable for use as a minimal detection criteria of these new, rapidly spreading, in the wild (but as yet unnamed) viruses.

The measurement of response times as currently designed into certification schemes such as those used by *NCSA*, and *SECURE COMPUTING Checkmark* is made to measure for protection models in which immediate response is unnecessary or unfeasible. An immune system model can meet the needs of the users in situations where Internet connectivity and viral increase will cause a two or three month time lag in documenting certification measurement

to be clearly unacceptable; the current certification model cannot effectively measure this response time. Automated updates, push technology enabled updates, and updates available via File Transfer Protocol sites and Bulletin Board Systems may be documented in the cooperative *CLEF*/Developer effort that is part of the *ITSEC Anti-Virus Working Group* model. However, while this current certification model has the potential to provide some measurement of vendor response time and reliability using these, in the future when new viruses spread worldwide in a matter of days, hours, or even minutes, response time problems will render even these approaches too slow. Indeed, we believe some of these approaches are already outdated.

The testing of an immune system model will a high degree of competency and technical expertise with not only anti-virus software and virus sample replication, but with the Internet and networked systems in general.

Conclusions

We have looked briefly at the history of the term "In the Wild" and how this developed into *The WildList*, the *de facto* standard for building test-sets made up of those viruses in the wild. We have then examined three certification schemes based upon *The WildList*, and show that only one, *ITSEC*, appears to be constructed in such a way as to measure the ability of a vendor to track and match the current threat: the others are chiefly based upon *The WildList*, and suffer greatly due to the rapidly changing threat. In one case, we illustrated how a certified product might not even be able to detect viruses in the wild which were spreading 6 months prior to the current date.

We considered an alternate way of classifying viruses for certification purposes, and discovered that although the number of viruses is rising steadily, the actual threat posed by computer viruses to computers varies as a function of time. We have highlighted the importance of measuring a developer's ability to quickly respond to new viruses and supply updates in the field. In particular, we note that in the case of an e-mail-aware or Internet-aware virus, even automated signature distribution may be too slow to be of much practical help. The computer-computer interactions which are becoming more and more the models of the ways in which we conduct business on the Internet are rendering manual elements of viral isolation, sample capture, replication, and analysis too slow - only techniques such as *IBM's* immune system approach offer the type of response time needed to adequately protect from such a virus.

This has serious implications for those involved in the certification of anti-virus software. Tests based upon *The WildList* measure the ability of a product to protect the user far better than Zoo based tests. However, we question the long-term usefulness of *WildList*-based certification schemes, especially in light of the turnaround and maintenance time of certification. While we acknowledge *The WildList* to be much improved with definite scientific and practical value, we feel certifications based upon *The WildList* represent the bare minimum in terms of protection – their presence alone is insufficient to guarantee the protection of your company.

Acknowledgments

The author would like to thank Richard A. Ford and Steve R. White, IBM TJ Watson Research Center, for suggestions, corrections and general enlightenment.

Bibliography

1. Real World Anti-Virus Reviews and Evaluations-the Current State of Affairs. Sarah Gordon and Richard Ford. Proceedings 19th National Information Systems Security Conference. Baltimore Maryland. October 1996. pp. 526-535
2. David Chess. Private e-mail conversation. Used with permission.
3. Alan Solomon. Private e-mail conversation. Used with permission.
4. Roger Riordan. Private e-mail conversation. Used with permission.
5. Letters to the Editor. Virus Bulletin. July 1991
6. Measuring Computer Virus Prevalence. Jeffrey O. Kephart and Steve R. White. Proceedings of the Second International Virus Bulletin Conference. Edinburgh, Scotland, September 2-3, 1992, pp. 9-28.
7. In [1]
8. *The WildList.* Joe Wells. http://www.av.ibm.com
9. In [6]

10. Real World Anti-virus Reviews and Evaluations - the Current State of Affairs. Presentation. Sarah Gordon. 19th National Information Systems Security Conference. National Institute of Standards and Technology, National Computer Security Center. Baltimore Maryland. October 1996.
11. In [1]
12. A Perspective of Evaluation in the UK versus the US. Alan Borrett. Proceedings 18th National Information Systems Security Conference. Baltimore, Maryland. 1995. pp.
13. What A Winword Concept. Sarah Gordon. Virus Bulletin. September 1995.
14. Excel Yourself. Sarah Gordon. Virus Bulletin. September 1996.
15. In [8]
16. In [11]
17. The Administrators' Guide to Macro Viruses. International Virus Prevention Conference. Presentation. Richard Ford. Arlington, Virginia. 1997.
18. Virus Exchange BBS: A Legal Crime? Sarah Gordon. American Association for the Advancement of Science. Conference on Computer and Network Use and Abuse. Irvine, Califomia. 1993.
19. Technologically Enabled Crime: Shifting Paradigms for the Year 2000. Sarah Gordon. Computers and Security. October 1995. pp391-402.
20. Computer Viruses: A Brief Overview. Carrie France. August 1996. http://www.academic.marist.edu/papers/france/paper.htm
21. David Chess. Private Communication. Used with Permission.
22. Biologically Inspired Defenses Against Computer Viruses. Jeffrey Kephart, Gregory Sorkin, William Arnold, David Chess, Gerald Tesauro and Steve White. Proc. Int'l J. Conf. on AI (IJCAI-95), Morgan Kaufmann, San Francisco, 1995, pp. 985-996.
23. *ITSEC* Anti-Virus Working Group. A-V Product Standard Functionality Class F-AVIR. Draft Issue 10. March 1997.
24. Ethical Implications and Impacts of Anti-virus Research. Sarah Gordon. In Progress.
25. A Biologically Inspired Immune System for Computers. Jeffrey O. Kephart . High Integrity Computing Laboratory . IBM Thomas J. Watson Research Center. Published in Artificial Life IV. Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems. MIT Press. Cambridge, Massachusetts. 1994. pp. 130-139.

What is Wild?

Sarah Gordon
IBM TJ Watson Research Center

P.O. Box 704
Yorktown Heights, NY 10598

Prepared for the 20th National Information Systems Security Conference

symantec.

Cyberterrorism?

by Sarah Gordon

Senior Research Fellow
Symantec Security Response

and Richard Ford, Ph.D.
Independent Consultant

INSIDE

› The terrorism matrix
› Pure cyberterrorism
› Computers — the weapons of the cyberterrorist
› Defending against the new terrorism

Contents

Abstract 3

Introduction 3

The terrorism matrix 5

- Perpetrator 6
- Place 6
- Action 6
- Tool 7
- Target 7
- Affiliation 8
- Motivation 8

Pure cyberterrorism 8

Terrorism as theater? 8

The new terrorism 9

Computers — the weapons of the cyberterrorist 9

- Future research 9

Defending against the new terrorism 11

- Deterrence 11
- Criminal justice 11
- Enhanced defense 12
- Negotiations 12

Conclusion 13

References 15

About the Authors 16

Abstract

The term cyberterrorism is becoming increasingly common in the popular culture, yet a solid definition of the word seems hard to come by. While the phrase is loosely defined, there is a large amount of subjectivity in what exactly constitutes cyberterrorism. In the aftermath of the September 11th attacks, this is somewhat disconcerting.

In an attempt to define cyberterrorism more logically, a study is made of definitions and attributes of terrorism and terrorist events. From these attributes a list of attributes for traditional terrorism is developed. This attribute list is then examined in detail with the addition of the computer and the Internet considered for each attribute. Using this methodology, the online world and terrorism is synthesized to produce a broader but more useful assessment of the potential impact of computer-savvy terrorists. Most importantly, the concept of 'traditional' cyberterrorism, which features the computer as the target or the tool is determined to be only a limited part of the true risk faced.

Finally, the authors discuss the impact this new view of cyberterrorism has on the way in which one should build one's defenses. In particular, the breadth of the issue poses significant questions for those who argue for vertical solutions to what is certainly a horizontal problem. Thus, the validity of special cyberterrorism task forces that are disconnected or loosely connected with other agencies responsible for fighting the general problem of terrorism is questioned, and a broader, more inclusive method suggested. Keywords: cyberterrorism, terrorism, computer security

Introduction

If you ask 10 people what 'cyberterrorism' is, you will get at least nine different answers! When those 10 people are computer security experts, whose task it is to create various forms of protection against 'cyberterrorism', this discrepancy moves from comedic to rather worrisome. When these 10 people represent varied factions of the governmental agencies tasked with protecting our national infrastructure and assets, it becomes a critical issue. However, given the lack of documented scientific support to incorporate various aspects of computer-related crime into the genre 'cyberterrorism', this situation should not be surprising.

Despite copious media attention, there is no consensus methodology by which various actions may be placed under the nomenclature 'cyberterrorism', yet the term clearly exists in common usage. The term, first coined in the 1980s by Barry Collin (Collin, 1997), has blossomed in the last several years: "Protect yourself from the cyberterrorist"; "Insure yourself against cyberterrorism"; "Funding forthcoming to fight cyberterrorism" (Hamblen, 1999; Luening, 2000).

All of these sound nice, but the reality is that the reader, solution provider, or defender is often left to his own devices as to what the term actually means and thus what solutions should be created (or implemented). *When a government's or corporation's entire infrastructure may be at stake, subjectivity is useful but may not be the best evaluative tool.*

At the same time, research of this phenomenon shows that cyberterrorism cannot easily be defined. This creates a Catch-22 situation: the thing cannot be defined — yet without defining it, one cannot 'know' what it is one is fighting and hence come up with a good solution. Furthermore, even when there is an operational agreement on terms, if an attack/security event does not fit into one of the (often narrowly defined) categories, funding (and consequently investigation or technical remedy) may not be forthcoming.

For example, recently terrorists used a computer in Delray Beach, Florida to make their travel plans and purchase tickets, as well as using public library computers in the same town (Holland, 2001). How large the role computers played in the organization and execution of the attacks is, at this point, unclear, but the conclusion is obvious: computers and, in particular, the Internet, played a key role in the execution of the September 11th attacks. This concept is critical in evaluating the true problem we face in the virtual world: the use of computers in terrorist acts. While there are possible technical solutions that would have made this particular scenario more difficult, this task does not currently fall under the auspices of any government agency tasked with fighting cyberterrorism. Furthermore, as each of the actions cited above was not necessarily illegal prior to the attack, detection and prevention is made all the more difficult.

The most widely cited paper on the issue of Cyberterrorism is Denning's Testimony before the Special Oversight Panel on Terrorism (Denning, 2000). Here, she makes the following statement:

> *Cyberterrorism is the convergence of terrorism and cyberspace. It is generally understood to mean unlawful attacks and threats of attack against computers, networks, and the information stored therein when done to intimidate or coerce a government or its people in furtherance of political or social objectives. Further, to qualify as cyberterrorism, an attack should result in violence against persons or property, or at least cause enough harm to generate fear. Attacks that lead to death or bodily injury, explosions, plane crashes, water contamination, or severe economic loss would be examples. Serious attacks against critical infrastructures could be acts of cyberterrorism, depending on their impact. Attacks that disrupt nonessential services or that are mainly a costly nuisance would not.*

While Denning's definition is solid, it also raises some interesting issues. First, she points out that this definition is usually limited to issues where the attack is against "computers, networks, and the information storied therein", which we would argue is 'pure Cyberterrorism'. Indeed, we believe that the true impact of her opening statement ("the convergence of terrorism and cyberspace") is realized not only when the attack is launched against computers, but when many of the other factors and abilities of the virtual world are leveraged by the terrorist in order to complete his mission, whatever that may be. Thus, only one aspect of this convergence is generally considered in any discussion of cyberterrorism — an oversight that could be costly. Second, it is very different from the definition that appears to be operationally held by the media and the public at large.

Given the Augean task of attempting to define cyberterrorism, one way we might approach the task of understanding it is to throw away the very idea of defining it at all, and instead begin by breaking it down into its fundamental elements — each of which can be examined and used as a foundation for developing solutions which may be technical, legal, social, educational, or policy driven. After all, a word is meaningless in and of itself — it is only the relational concepts that the word conveys that imbue the utterance with meaning.

As 'cyberterrorism' relates to 'terrorism' a logical first step might be to look at the functional elements present in some operational definitions of 'terrorism'[1].

The United States Federal Bureau of Investigation (FBI) defines terrorism as, "The unlawful use of force or violence, committed by a group(s) of two or more individuals, against persons or property, to intimidate or coerce a government, the civilian population, or any segment thereof, in furtherance of political or social objectives." (FBI, 2002).

The United States Department of Defense (OOD) defines terrorism using a slightly broader brush, calling it "the unlawful use of, or threatened use, of force or violence against individuals or property, to coerce and intimidate governments or societies, often to achieve political, religious or ideological objectives" (OOD, 2002).

The United States Department of State (DOS) definition states that terrorism is "premeditated, politically motivated violence perpetrated against noncombatant targets by sub national groups or clandestine agents" (DOS, 2002).

These varied operational definitions exist as a function of the individual organizational roles and tasks which are assigned to employees/agents. Thus, as these roles and tasks vary, the concepts of terrorism continue to vary.

The terrorism matrix

When terrorism is examined in view of these definitions, there are some pervasive elements: people (or groups), locations (of perpetrators, facilitators, victims), methods/modes of action; tools, targets, affiliations, and motivations[2]. Examples are shown in Figure 1, using two groups designated as terrorist groups by the United States government: The Liberation Tigers of Tamil Eelam (LTTE) and the Aum.

	LTTE	AUM
Perpetrator	Group	Group
Place	Sri Lanka	Japan
Action	Threats/Violence	Violence
Tool	Kidnapping/Harassment	Nerve Gas
Target	Government Officials/Recruits	!=AUM
Affiliation	Actual/Claimed	Actual/Claimed
Motivation	Social/Political Change	World Domination

Figure 1: Terrorism matrix by group and attribute.

When we examine the elements in these categories in terms of the definitions provided by the government agencies, we see there is congruence between the terrorism event and the definitions used by the various agencies tasked with providing protection. This congruence is a good thing, as it results in people tasked with defense being able to determine that certain functional tasks (building blocks of modeling solutions) fit within the definitions used within their agencies/organizations.

For example, as mentioned above, the United States Department of State (DOS) defines terrorism as "premeditated, politically motivated violence perpetrated against noncombatant targets by sub national groups or clandestine agents". Thus, the activities of both of these groups fit the DOS criteria for 'terrorism'.

Integrating the computer into the matrix of their traditional terrorism introduces some interesting effects and problems, as we see when we consider two groups, the LTTE and Aum referenced in Figure 2. Note how the scope of 'terrorism' changes within each cell due to the addition of the computer.

1. It is worth noting that there are over 100 definitions; examining them all is beyond the scope of this paper.
2. Note that desired and actual outcome play a role as well. Future 'performance' in the theatre of terrorism is likely to be based to some degree on 'audience reaction'. This is integral to the discussion, but beyond the scope of this short paper.

	LTTE	AUM
Perpetrator	Group/Individual	Group/Individual
Place	Sri Lanka/London/Australia/ Worldwide	Japan/US/Worldwide
Action	Threats/Violence/Recruitment/ Education/Strategies	Violence/Recruitment/ Education/Strategies
Tool	Kidnapping/Harassment/ Propaganda/Education	Nerve Gas/Education
Target	Government Officials/Recruits	!=AUM
Affiliation	Actual/Claimed	Actual/Claimed
Motivation	Social/Political Change	World Domination

Figure 2: Matrix of terrorism with inclusion of the computer.

In this model, not all of the elements are congruent with functional tasks assigned to given agencies. Thus, 'terrorism' can take place within these same groups that is not within the scope of investigation, etc. This is clearly a major problem, and one that merits further investigation. Therefore, let us look very briefly at the various sorts of issues the inclusion of computers introduce to the concept of terrorism. This is obviously an extremely complex task; each area will be considered in depth in future research, and as part of the IFIP World Computer Congress workshop on Cyberterrorism (WCC, 2002).

PERPETRATOR

Interactions between human beings are complex; while the obvious solutions gravitate toward monitoring, we are concerned with virtualization of interactions, which can lead to relative anonymity and desensitization. Topics of interest include methods to measure and diminish the impact of computer-mediated interactions on potential recruits and the ability for defenders to use virtual identities to influence intra- and inter-group dynamics (dissension, 'behind the scenes' communication and destabilization).

PLACE

Location exists as an element, but is not a 'required' element in traditional terrorism in that an event does not have to occur in a particular location. Thus, whether an act is virtual/virtual, virtual/real world or real world/virtual is of interest only as factor in modeling solutions. In addition, the Internet has introduced globalization of the environments.

Actions that take place in virtual environments have demonstrably had real world consequences. An April Fool's Day hoax posted to Usenet demonstrated this when claims of the resignation of Canadian Finance Minister Paul Martin resulted in the decrease in value of the Canadian dollar (Reuters, 2002).

ACTION

In traditional scenarios, terrorist scenarios typically are violent or involve threats of violence. While there have been many studies of violence in the physical world, more research is called for in terms of 'violence' as a virtual phenomenon. Violence in virtual environments is a relatively new field, with many unanswered questions. These open issues include the psychological effects of traditional real-world violence portrayed in virtual environments, possible behavior modification resulting from

violence in virtual environments, physical trauma from virtual violence and the use of virtual violence in military training (Stone, 1993; Whiteback, 1993). However, despite the prevalence of traditional violence portrayed in virtual environments, 'cyberviolence' is still very much an unknown quantity. For example, destruction of someone's computer with a hammer constitutes a violent act. Should destruction of the data on that machine by a virus also be considered 'violence'? Perhaps violence should be considered in terms of hostile action, or threat thereof?

TOOL

There are an almost uncountable number of ways that the terrorist can use the computer as a tool. Facilitating identity theft, computer viruses, hacking, use of malware, destruction or manipulation of data all fall under this category.

These uses of the computer, when combined with 'computer as target' form the 'traditional' picture of cyberterrorism. These will be discussed in more detail later in the section Computers: The Weapon of the Cyberterrorist.

TARGET

There are a large number of potential targets that involve, either directly or indirectly, computers. Consider, for example, the impact of Personal Identity Theft. While the incidence of identity theft is comparatively low, the impact of theft upon the unfortunate soul whose ID is stolen can be large: terrorists could use the stolen identity to mask their work, carrying out certain operations under their target's name, not their own.

This would help evade detection by authorities, as well as potentially acting as a 'signal' that an identity or operation had been compromised. The Internet, especially the essentially useless authentication provided by email, provides the perfect breeding ground for identity theft.

Another interesting twist on this scenario is that of 'virtual' identity theft. For example, many users have multiple online personalities or profiles. Conceptually, there may be reasons why a terrorist would benefit from stealing a user's online identity. Attacks could be as trivial as exploiting trust relationships with other users when logged in as the stolen identity, to planting of Trojans etc., via 'trusted' email.

Similarly, the rise of online stock trading and stock message boards has created an environment in which it is possible to deliberately manipulate a stock price (perhaps via a stolen identity). A terrorist could use such techniques as a funding source, or even attempt to move the markets towards chaos. Thus, a well organized virtual attack upon a bank or corporation's stock rather than the bank or corporation itself, could in fact prove to be highly effective.

In the opinion of the authors, all of the attacks mentioned above are more likely to be successful when carried out against individual users or corporations rather than governments. However, governmental control currently relies heavily on the stability of the overall economy; thus economic destabilization is a viable attack against a government as well as the attacked third-party entity.

Using the terrorism matrix, effective solutions for computer as 'target' can be conceptualized and designed — but these will be useless overall unless problems (technical, social, legal) arising from the interaction of computer with every cell of the terrorism matrix is addressed. If I can buy a ticket for an unknown 'friend' in Bulgaria to fly to London and blow up the London Eye, antivirus software on the computer controlling the London Eye is of little relevance.

AFFILIATION

It is possible for a person to read all about a given cause and chat with proponents of the cause without ever leaving the safety of his or her own home. New recruits can thus become affiliated with a terrorist group, commit to carrying out given actions, all without ever actually coming into contact with another human being. At the same time, these loose affiliations can complicate investigations and confuse media reports. Additionally, the introduction of computing technology facilitates alliances between groups with similar agendas; this type of affiliation can result in strengthening of the individual organizations as they can immediately acquire access to the information resources of their allies.

MOTIVATION

Political, social, and economic changes are the motivations present in real-world terrorism. Combining a dependence on Internet-connected systems for banking and Ecommerce with the ability of anyone with a desire and readily available tool to disrupt these areas, results in a situation that is all too clear: unless steps are taken to significantly reduce risks, disaster is inevitable. Even with the best risk reduction, there are still likely to be problems.

> Pure cyberterrorism

The concept of 'pure' cyberterrorism — that is, terrorism activities that are carried out entirely (or primarily) — in the virtual world is an interesting one. The Internet provides many different ways of anonymously meeting with 'like minded' individuals in a (comparatively) safe way. Furthermore, a successful cyberterrorism event could require no more prerequisite than knowledge — something that is essentially free to the owner once acquired, and an asset that can be used over and over again. Thus, it would be possible that such an environment could facilitate the creation of entirely new terrorist groups — no monies would be required for actions, and members could organize themselves quickly and easily in the anonymity of cyberspace. This is very different from certain examples given above, where the computer can aid the task of the terrorist, but 'real' resources are still required to execute the plan. It is this pure cyberterrorism that most writers mean when they discuss the dangers posed by the cyberterrorist, and this compartmentalization poses a significant barrier to our ability to protect ourselves.

One question that has not been adequately addressed in the literature is what might this terrorism look like. At this time, there is much confusion, based largely upon lack of agreement in definitions. However, using 'traditional' terrorism models should help make the situation more suited to analysis, and this is certainly a topic for future research.

> Terrorism as theater?

Within the terrorism literature, a common metaphor is that of terrorist incidents as theater. Those concerned with terrorism and the media frequently find the staging of incidents, the publicity sought, and the manipulation of the audience primary themes in their analyses. To this end, WWW sites can bring publicity, and this is indeed a growing trend. Additionally, currently almost half of the 30 groups on the State Department's list of terrorist organizations have their own websites, which can be used to solicit money for their various causes or disseminate coded messages, either explicitly or steganographically.

The functional tasks of the group having a WWW presence may be distributed among several sites; it is relatively easy for a terrorist organization to solicit funds for operations via the WWW (the ultimate penetration of Ecommerce?), promote their cause, as well as recruit would-be operatives while maintaining somewhat of a perceptual distance between the tasks. Finally, the relative anonymity provided to those accessing information via the WWW also helps distance those sympathetic with the cause from those actively fighting for the cause in ways that may be objectionable to the sympathizers.

The new terrorism

New terrorist organizations are highly funded, technologically articulate groups capable of inflicting devastating damage to a wide range of targets. While most published work in the computer industry has focused on the impact of the computer as target (pure cyberterrorism) it is our belief that the real danger posed by the synthesis of computers and terrorism is not only the insertion of computer as target in the terrorism matrix, but in many of the other areas, too.

The current narrowness of focus poses a significant risk to US infrastructure. By being too concerned about one particular part of the matrix, we are apt to let our guard down in areas which may be more critical. A forward-looking approach to terrorism that involves computers is highly contextual in its basis. Traditional antiterrorism defenses must be deployed, but these countermeasures must fully take into account many of the virtual factors that we have outlined in this paper.

If the events of September 11th teach us one thing, it is that we should always consider the 'big picture' of the overall terrorist threat, rather than view one aspect in isolation. The 'cyber' aspects of the puzzle must be woven throughout the picture, not simply confined to one cell. To view a problem with too narrow a perspective is to invite anarchy into our lives.

Computers — the weapons of the cyberterrorist

Following on from the discussions above, it becomes obvious that the most likely 'weapon' of the cyberterrorist is the computer. Thus, one might ask, are we arguing that one should restrict access to computers, just as access to explosives is restricted? Not quite, but close. We believe that the stockpile of connected computers needs to be protected. There are many laws that define how one should protect a firearm from illegal/dangerous use. The mandatory use of trigger locks, though controversial, has been put forward to prevent danger should the gun end up in the wrong hands. Similarly, powerful explosives like C4 are not simply sold over the counter at the corner store.

Explosives and guns are certainly not entirely analogous to computers. A better analogy might stem from the concept of an 'attractive nuisance'. For example, a homeowner shares some responsibility for injury caused by a pool on his property — it is deemed an attractive nuisance, and as such, the innocent should be prevented from simply being attracted and harmed.

Thus, there are many instances of laws which already discuss damage done by/to a third party from the intentional/unintentional misuse of a piece of corporate/personal property. The application of these laws or the definition of 'misuse' with respect to computers seems unclear. However, there is a need for clear laws and standards which require operators of large networks of Internet-connected computers to exercise appropriate due diligence in their upkeep and security.

To this end, we believe that there is an urgent need for definition of a minimum standard of security for computer networks. The definition of such a standard has far reaching implications not only for the usability of America's technology foundation, but the security of corporations and indeed of the nation itself. By formalizing an industry best practice guideline, companies will have a clear understanding of what must be carried out.

Clearly, such a guideline is a moving target, but its inception would allow the structuring of a valid and robust posture against both terrorist threats and other hostile entities. Such a set of minimum standards would have to be easily and affordably supported by the security/application vendors themselves, rather than relying on individual users needs/requirements to drive the best practice guidelines.

This is not exactly a novel concept. International standards have been developed in other areas where safety and security are a concern. Consider the airline industry. There are international guidelines for airport safety; in cases where these standards are not met, consequences range from warnings to prohibited travel. The needs for such changes, and how a due diligence standard could be created are subjects of future research. However, it seems clear that such standards are urgently needed.

FUTURE RESEARCH

Certainly there are many unanswered questions. Most people, governments included, consider cyberterrorism primarily as the premeditated, politically motivated attack against information, computer systems, computer programs, and data by sub national groups or clandestine agents.

However, as we have seen, the real impact of the computer on the terrorism matrix is considerably wider. By limiting our understanding of cyberterrorism to the traditional 'computer as target' viewpoint, we leave our nation open to attacks that rely on the computer for other aspects of the operation.

Even when considering the purely virtual impact of cyberterrorism, the approach is not adequately thought out. For example, consider an act that incorporated a desire for political change with the release of an otherwise benign computer virus within which an antigovernment message is embedded. For example, if the Melissa virus had contained the message "The Clinton regime must be defeated", would it have been the act of a terrorist instead of a misguided computer programmer — and would the ultimate punishment really fit the crime if that programmer were meted out the same punishment as the terrorists responsible for blowing up a US embassy?

What role does incitement to violence play? A swastika emblazed on the WWW site of UK politician John Major may constitute some violation of a law, but probably does not constitute terrorism. But what if swastikas were digitally painted on the WWW sites of every Jewish organization in the country? What if a message was included inciting people to violence against their Jewish neighbors? Would these acts fall under the domain of 'using violence'? What if these images and messages were put there by a known terrorist organization? Would the act take on the characteristic of the perpetrator? Would these acts be hate crimes or cyberterrorism? Whence falls 'jurisdiction'?

Given the lack of physical boundaries in the virtual community, does a group's physical location have any bearing on whether or not they may be considered a sub-national group? What is a 'national group' in cyberspace anyway? Which government agency deals with *that?*

What constitutes combatant targets in virtual environments? Consider the 1998 response by the Pentagon to civilian target computers as a response to Floodnet protests (McKay, 1998). Is the system that automatically strikes back considered combatant? Are its owners moved from 'non-combatant' to 'combatant' based on an auto-response? Is the response perhaps engaging in 'violence'?

Some claim "terrorists and activists have bombed more than 600 computer facilities". What specific components may be considered an element of a cyber system; what differentiates these incidents from conventional terrorism? Physical property, civil disorder and economic harm are easily understood in the physical world; however, are there virtual equivalents that could lead to a broadening of the concept of cyberterrorism?

Defending against the new terrorism

Defending against terrorism where a computer or the Internet plays an important part in the terrorism matrix is very similar to defending against terrorism that does not. The regular practices (deterrence, law, defense, negotiations, diplomacy, etc.) are still effective, except that the scope of certain elements is expanded. For example, traditional strikes against military bases, targeting of key leaders, and collective punishment have been effective in traditional terrorism (Whitelaw, 1998) and certainly have potential for dealing with some aspects of cyberterrorism. These techniques are often presented, and can be to be updated to include their 'virtual' counterparts. It should be noted, however, that differences in international law and culture could make this process a complex task.

Crenshaw (Crenshaw, 1999) presented here at length, examines a summary of traditional counter-terrorist techniques:

DETERRENCE

Governments can use their coercive capacity to make terrorism too costly for those who seek to use it. They can do this by military strikes against terrorist bases, assassinations of key leaders, collective punishment, or other methods. There are several drawbacks to this approach, however. On the one hand, it can lead to unacceptable human rights violations. In addition, groups may not come to government attention until movements are so well developed that efforts to contain them through deterrent methods are insufficient.

CRIMINAL JUSTICE

Governments can treat terrorism primarily as a crime and therefore pursue the extradition, prosecution, and incarceration of suspects. One drawback to this approach is that the prosecution of terrorists in a court of law can compromise government efforts to gather intelligence on terrorist organizations. In addition, criminal justice efforts (like deterrent efforts) are deployed mostly after terrorists have struck, meaning that significant damage and loss of life may have already occurred.

ENHANCED DEFENSE

Governments can make targets harder to attack, and they can use intelligence capabilities to gain advance knowledge of when attacks may take place. As targets are hardened, however, some terrorist groups may shift their sights to softer targets. An example is the targeting of US embassies in Kenya and Tanzania in August 1998 by truck bombs. Although the attacks are believed to have been coordinated by individuals with Middle Eastern ties, targets in Africa were chosen because of their relatively lax security compared with targets in the Middle East.

NEGOTIATIONS

Governments can elect to enter into negotiations with terrorist groups and make concessions in exchange for the groups' renunciation of violence. While governments are often reluctant to do so at the beginning of terror campaigns, negotiations may be the only way to resolve some long-standing disputes.

For example, data gathering and monitoring operations of terrorist communications has typically applied to signal intelligence and fieldwork. In a virtual environment, the ability to gather information from various sources is eminently achievable in a somewhat automated manner. Specific groups can be watched easily, and computers are comparatively simple to 'bug'. All contacts that a particular user interacts with could then be tracked, and the network of communication mapped. Furthermore, much of this surveillance can be carried out over the very same network that the terrorists intend to use to facilitate their plot.

This extension, however, must be carried out with care. Consider, for example, the original US export regulations on the export of 'strong encryption' (ITAR). Under such regulations, certain encryption products were classified as munitions. While ITAR has since been replaced, the revamped 'Export Administration Regulations' (DOC, 2002), while somewhat more relaxed, continue to blacklist several countries from receiving encryption products, despite the fact that strong encryption technology is freely available via the Internet. While this law seems to be aimed at preventing the use of strong encryption by other potentially hostile governments and terrorist entities, strong encryption algorithms and implementations remain trivially available to pretty much anyone.

This classification of knowledge as munitions seems to be the ultimate (and flawed) extension of traditional anti-terrorist tactics into the virtual realm. Clearly, it is not sufficient to quickly draw analogies that are not, in fact, correct. A far better approach is to carefully consider the impact of the computer in the different cells of the terrorism matrix. For example, banning the export of encryption from just America is akin to banning the sale of C4 only on weekdays — the asset would be hardly even an inconvenience to the would be terrorist. A far better solution is to consider the safeguard in the context of the virtual world. When examined in this aspect, for example, it is reasonably clear that the original classification of encryption products as munitions is not likely to be effective. Similarly, while the use of export grade encryption can (and has) resulted in the ability of officials to read some terrorist communiqués, a restrictive "export to here, not here" ban is unlikely to succeed in any meaningful way.

A forward-looking approach to terrorism that involves computers is therefore highly contextual in its basis. Traditional antiterrorism defenses must be deployed, but these countermeasures must fully take into account the virtual factors that we have outlined in this paper.

> Conclusion

The Internet was developed primarily as an unregulated, open architecture. Not only are we observing a predictable backlash to the 'corporatization' of the network, where the tools of destruction can easily be placed in the hands of the dissatisfied or malevolent people, we must also deal with the fact that the infrastructure is ideally suited to criminal activities. Some of these activities are being promoted as cyberterrorism; however, the loose use of the term is actually undermining the defense capabilities of the very corporations and governments who are at risk.

Events can be analyzed in terms of their critical factors, and only if these factors all exist can the event legitimately be called terrorism. However, that does not mean if all these factors do not exist that a corporation is 'safe'. Unfortunately, corporations are built around the premise that people will do the right thing. The fact, as we have seen, is that this is not necessarily the case.

We do not use the term 'ice pick terrorism' to define bombings of ice-pick factories, nor would we use it to define terrorism carried out with ice picks. Thus, we question the use of the term cyberterrorism to describe just any sort of threat or crime carried out with or against computers in general. At the same time, those who do insist on treating only 'pure cyberterrorism' as cyberterrorism are completely missing the true threat posed by the addition of acts in the virtual world to the terrorists' playbook. Finally, the nascent danger in the term cyberterrorism is that cyberterrorism will somehow be dealt with separately to regular terrorism. This artificial fragmentation of our defenses is likely to provide the terrorist with a significant advantage in any campaign against a nation state, and is to be avoided at all costs.

This brings us to the final point of this study: turning the tables on terrorism. As we have shown, computers can play an enormous role in terrorism. At the same time they can provide perhaps our biggest defense against terrorism if used to our advantage. However, just like as we need to understand the integration of computers with terrorism, we must examine how computers can assist in defense broadly.

This begins with the re-examination of basic beliefs about 'cyberterrorism' which must take place within academia, industry, government and defense sectors. This re-examination is, however, only the first step in combating terrorism.

Information at each level of analysis must be shared, collated and redistributed across federal, state and local government boundaries, as well as amongst industry and academia, and in some cases, the private citizenry.

Obviously, this type of endeavor is information technology intensive. In the United States alone there are over 87 000 different jurisdictions (HLS, 2002); combined with information from industry, academia and the private citizenry, this amount will increase many times over. Fortunately, one of the things computers are good at is processing information. By developing an information technology architecture that would sort, correlate and facilitate the most effective use of that information, information could be shared in a timely manner amongst these groups, some of which historically have had little, if any, communication.

While specification of design or administration of such a project is beyond the scope of this paper, we believe that using a publish and subscribe model, with a meta-information standard such as XML, data from law enforcement, government, defense and industry could be analyzed, correlated, filtered and redistributed quickly to those who need it most. This cross-disciplinary sharing of information could help practitioners create complementary defensive solutions and policies, building on shared expertise and innovation.

Additionally, a well-designed technical solution can circumvent some of the cultural problems inherent in cross-sector information sharing, by eliminating the need for the actual data to do the correlation. Some technologies have been developed which would appear to lend themselves particularly well to this sort of implementation, but practical tests are required before any conclusion can be reached (Legion, 2002).

Aside from the role of computers in defense, we must attempt to re-educate policy makers, defusing the latent danger of vertical 'cyberterrorism' defenses and replacing them with a well-rounded, integrated approach to a problem that is extremely broad. From a corporate and governmental perspective this requires a careful examination of the 'messaging' that is broadcast. How do we portray the fusion of computers with terrorism? Can the messaging be made more productive so that we can shape the mindset of our audience to one that is synergistic with a broad view of cyberterrorism?

Finally, it is impossible to neglect to mention the fact that the rapid increase in connectivity and the ultimate frailty of our national IT infrastructure coupled with the astonishing homogeneity of our computing base is a matter of grave concern. Continued focus must be put on increasing the public demand for computer security as well as the corporate awareness of the issue: whereas security flaws in widely used applications were once perceived as personal risks, we must begin to recognize the potentially global consequences of such issues in balance with the more general problems posed by the integration of computing with terrorism.

The lack of understanding of cyberterrorism, and the overall insecurity of America's networks have allowed a situation to develop which is not in the best interests of the country or computer users. The need to protect computing resources, making the job of a cyberterrorist more difficult is obvious. However, this can only be accomplished by re-examining commonly held beliefs about the very nature of computer systems and of cyberterrorism itself.

References

Collin, B., 1997. *The Future of Cyberterrorism*, Crime and Justice International, March 1997, pp.15-18.

Crenshaw, M . 1999. *How Terrorism Ends.* US Institute of Peace working group report, May 1999.

Denning, D., *"Cyberterrorism"*, Testimony before the Special Oversight Panel of Terrorism Committee on Armed Services, US House of Representatives, 23 May 2000. (http://www.cs.georgetown.edu/~denning/infosec/cyberter ror.html)

DOC, 2002. *US Department of Commerce, Export Administration Regulations* (EAR), 15 C.F.R. Parts 730-774. Sections 740.13, 740.17 and 742.15 are the principal references for the export of encryption items.

DOD, 2002. *Department of Defense Education Activity Internal Physical Security.* Department of Oefense. DoDEA Regulation 4700.2

DOS, 2002. United States Code. Title 22, Section 2656f.

F8I, 2002. *Code of Federal Regulations.* 28 CFR. Section 0.85 on Judicial Administration. July 2001.

Hamblen, *M. Clinton commits $1.46B to fight cyberterrorism* http://www.cnn.com/TECH/computing/9901/26/clinton.idg , 26 January 1999.

HLS, 2002. *National Strategy for Homeland Security.* Office of Homeland Security. July 2002.

Holland, J. 2001. *Investigators look into how library computers may have linked terrorists.* South Florida Sun-Sentineel. Miami, FL.

Legion, 2002. http://legion.virginia.edu Retrieved from the World Wide Web, August 2002.

Luening, E. 2000. *Clinton launches plan to protect IT infrastructure.* CNET, 7 January 2000.

McKay, N. 1998. *Pentagon Deflects Web Assault.* Wired, September 1998.

Reuters, 2000. *Canadian dollar in for a ride with Martin Firing.* http://biz.yahoo.com/rf/020602/canada_economy_dollar_2 .html.

Stone, V. 1993. *Social interaction and social development in virtual environments* . Teleoperators and Virtual Environments, Vol. 2. pp. 153-161.

Whiteback, C. 1993. *Virtual environments: Ethical issues and significant confusions.* Teleoperators and Virtual Environments, Vol. 2. pp. 147-152.

Whitelaw, K. 1998. *Terrorists on the Web: Electronic 'Safe Haven'.* US News & World Report, Vol.124, p. 46.

WCC, 2002. *IFIP World Computer Congress Workshop.* IFIP World Computer Congress. Montreal, Quebec, Canada. August 2002.

This article first appeared in 'Elsevier Science Computers and Security Journal'.
For more information please visit www.compseconline.com.

About the Authors

Sarah is Senior Research Fellow at Symantec Security Response. Her current research areas include testing and standards for antivirus and security software, privacy issues, cyberterrorism and psychological aspects of human/computer interaction.

She has been featured in diverse publications such as IEEE Monitor, The Wall Street Journal, and Time Digital, and profiled by PBS, ITN, and CNN International. Her work has appeared in publications such as Information Security News and Virus Bulletin; she has won several awards for her work in technology. She is a highly sought-after speaker, having presented at conferences ranging from DEFCON to Govsec.

She was recently appointed to the Editorial Board for Elsevier Science Computers and Security Journal. She is on the Advisory Board of Virus Bulletin, and is co-founder and board member of The WildList Organization International. She is Technical Director of The European Institute for Computer Antivirus Research – where she also serves on the Board of Directors and Conference Program Committee. She is a member of SRI's cyberadversary working group.

Sarah was responsible for security testing and recommendation for The United Nations, and participates in various initiatives for Homeland Security and Infrastructure Protection. She was chosen to represent the security industry in "Facts on File: Careers for Kids who Like Adventure", and is a reviewer for various technical security book publishers including Wiley publications. Her work in ethics, technology and profiling computer criminals is required coursework in various academic information security programs. She is committed to excellence in information security education, guest lecturing at Universities world-wide on topics ranging from virus writers and hackers to the truth about cyberterrorism.

Sarah graduated from Indiana University with special projects in both UNIX system security and ethical issues in technology. She is a member of the American Association for the Advancement of Science, The American Counseling Association, and the Association for Family Therapy and Systemic Practice in the UK. Prior to joining Symantec, she worked with the Massively Distributed Systems Group at IBM's Thomas J. Watson Research Laboratory in New York in the AntiVirus Research and Development Team. She may be reached at sgordon@symantec.com.

Dr. Richard Ford is one of the nation's top Internet architecture experts and an internationally acknowledged computer security specialist. Ford has lectured worldwide on the subject of computer security, and holds editorial positions for several virus and security related journals. He holds one patent and is widely published in the areas of both computer security and physics. Dr. Ford has served as Director of Research for the National Computer Security Association. He also spent two years at IBM Research's prestigious T.J. Watson Research Center, working in the Massively Distributed Systems group's High Integrity Computing Laboratory. Before being appointed Chief Technology Officer for Cenetec Ventures, Dr. Ford was the Director of Technology at Verio in Boca Raton. There he served as senior architect for the world's largest web hosting platform and was also responsible for the security of more than 200,000 web sites. He is currently working as a consultant on various projects related to computer security and e-commerce.

Dr. Ford was educated at Oxford in his native England and holds a Bachelors Degree and a Masters in Physics from that university. He went on to complete his doctorate at Oxford in Low Temperature Semiconductor Physics. He began work immediately after as an executive editor of Virus Bulletin, the world's foremost technical publication on computer viruses and malicious software.

Now an independent consultant, Ford resides in Boca Raton with his wife Sarah.

WORLD HEADQUARTERS

20330 Stevens Creek Blvd.
Cupertino, CA 95014 U.S.A.
408.517.8000
800.721.3934

www.symantec.com

For Product information in the U.S. call toll-free 800.745.6054

Symantec has worldwide operations in 38 countries. For specific country offices and contact numbers please visit our Web site.

symantec.

Privacy: A Study of Attitudes and Behaviors in US, UK and EU Information Security Professionals

By Sarah Gordon
Senior Research Fellow
Symantec Security Response

INSIDE

› What is Privacy?
› Inadvertent Disclosure
› Malicious Disclosure
› Technical Responses to Privacy Threats

Contents

Abstract 3

What is Privacy? 4

Culture, Gender and Privacy 5

Technical Aspects of Privacy 6

Inadvertent Disclosure 7

- Web Usage - Cookies 7
- Web Usage – Privacy Policies 7
- Email – Spam Tracking Pixels 7
- Downloads – End User License Agreements 8

Malicious Disclosure 8

- Password-Stealing Trojans 8
- Spyware 8
- Remote Access Trojans 9
- Computer Viruses 9
- Blended Threats 9

Technical Responses to Privacy Threats 10

Study Goals 11

Methodology 11

Responses Summary 12

Analysis 13

Cognition 14

Conclusion 14

References 15

About the Author 16

Abstract

As technology continues to modify the ways in which information of all types is stored, analyzed and exchanged, concerns related to privacy are growing. At the same time, the very concept of privacy is highly subjective, varying culturally as well as organizationally. In this presentation some of the cultural and organizational aspects of privacy will be examined, and some Internet-related threats to privacy discussed. Then, new survey data from our study of user behavior and technical facilitators of privacy will be presented. The study focuses on users' attitudes toward privacy and their responses to some globally applicable privacy-related threats. The data show some unexpected results, which will be interpreted by application of several well-known psychological models to the user behavior. Finally, the need for further work in the field is highlighted, and suggestions for further research provided.

What is Privacy?

Privacy is a relatively new concept. While the word "privacy" first appeared in the 15th century, the meaning most closely related to how the word is used today did not emerge for another four hundred years. As shown by the following varied views of privacy, privacy is comprised conceptually of both private and public spaces; it is context dependent and varies from person to person.

For some, privacy is exercising control over the information about themselves, or their family, that others have access to [Chess, 2003; Stefnisson, 2003]. For others, privacy is only doing things that have been expressly permitted with personal information [Whalley, 2003]. Privacy is sometimes seen as extending from information about a person to information about what a person does: for example, [Raiu, 2003] states "privacy is all data I'm working with and which shouldn't be available to just anyone is part of my (personal) privacy, and that includes e-mails, malware collections, or program sources." Some believe privacy consists of preventing others from knowing things which they know, but do not wish them to know; thus, it could be related to any type of information - not just information about oneself [Shipp, 2003]. For some, privacy extends to a right to prevent being contacted or approached by parties without consent [Kaminsky, 2003]; many people's perspective on UCE (Unsolicited Commercial Email, or Spam) illustrates this view of privacy.

In terms of popular usage, dictionaries tend to provide an excellent insight into the way a word is commonly used [Websters, 2003] defines privacy as "the quality or state of being apart from company or observation; freedom from unauthorized intrusion," and does not specify whether this relates to people or data. [OED, 2003] states privacy is "The state or condition of being alone, undisturbed, or free from public attention, as a matter of choice or right; freedom from interference or intrusion." Given these varied definitions of privacy, it is important to define the aspect of privacy that this study attempts to investigate. Based upon the explosion of Internet access, it seems meaningful for the purpose of this paper to operationally define privacy as the control over the disclosure of information about one's self or personal transactions.

Culture, Gender and Privacy

The concept of privacy is ever evolving; today individuals face a wide variety of privacy concerns. One of these concerns is how companies or organizations handle private, or personal, information provided by the individual. There are some cultural differences in the amount of trust we put in others to handle this type of personal information. For example, a 1999 study by [IBM, 1999] found that Americans slightly placed more confidence in companies handling of their personal information than did people from Germany or the United Kingdom. However, there are also differences in how we perceive what information should be publicly available in the first place. In Sweden, for example, some information from tax returns is public information, whereas in some other countries, this would be considered a gross violation of privacy "rights"[1]. Many other cultural differences in privacy exist. For example, homes in Arabian society are constructed so that the residents of the house cannot see their neighbors from any part of the house, thus insuring the privacy of the neighbors [Al-Sabt, 1995]. Interestingly, this cultural expectation for privacy of one's neighbors rests primarily not upon the neighbors, but upon the one building the house that might allow for inadvertent viewing of the neighbors. [Fullbright, 2003] comments on Japanese privacy norms: "Americans frequently comment on the different sense of privacy, both physical and psychological, between Japanese and Americans.... In the bank when conducting a transaction or using the cash machine, it may be disconcerting to find someone standing right behind you ... in the typical hospital or dentist's office the doctor will examine the patient not in an enclosed private office but frequently in a curtained-off area."

Gender also appears to play a role in some of the issues related to privacy. Many, if not most, studies on gender and privacy have focused on behaviors that sexually objectify women such as the use of skirt-cams, pretexting, familial abuse and societally imposed modesty [Allen, 2000; Marx, 2003]. A recent study by Information Technology Association of America found that women felt half as safe as men online, in several areas including the control over disclosure of their private information [ITAA, 2003]. One thing is clear from the existing research: women and men differ in what they believe about privacy, what they expect in terms of privacy, and in what they are willing to do to protect their privacy.

[1]While in the U.S, the Freedom of Information (1966) and Canada's Privacy Act (1985) were both established relatively recently, Sweden's Freedom of the Press laws were established in the early 1700s, and set a precedent for conceptualizing "private information".

Technical Aspects of Privacy

Aside from the different cultural expectations and definitions of privacy, one of the reasons why the concept of privacy has become so important is the ability of technology to provide for massive and fundamental changes in terms of abuses of privacy. As a trivial example, consider the "contemporary" issues of privacy from 100 or even 50 years ago. Many transactions were carried out in cash, essentially making them untraceable. Public records, if they existed at all, had to be manually searched. The process of inference (determining classified information from a large number of unclassified records) was difficult and time-consuming.

A quick comparison to the interconnected world of today provides an astonishing contrast. While we have always offered up personal information about ourselves (for example, when applying for insurance or benefits, obtaining medical services, filing tax returns, applying for employment, seeking credit, getting a mortgage, etc.), this information was relatively secure. However, the advent of large databases maintained by companies that specialize in collecting huge numbers of public records allows for the trivial monitoring and investigation of an individual. Data mining makes the process of inference cheap and easy, and the move from cash to credit cards, phones to cellular phones and paper mail to email make the task of investigating a particular citizen easier.

Although many facets of the impact technology can have on privacy are well explored by experts in law and public policy, there are some gaps in research to date. As we examine some of the previously unexplored issues, we will first consider inadvertent disclosure of private information - the "leakage" of information that the user either explicitly or implicitly allows whilst using his computer. Following this, we will explore malicious disclosure of information, via various forms of malicious code.

Inadvertent Disclosure

There are many different ways in which a user can inadvertently compromise their privacy. For example, even the simple actions of browsing the web, downloading software or purchasing software online can impact user privacy. In this section, some of these disclosures are examined, in order to illustrate the types of risk faced.

WEB USAGE – COOKIES

A cookie is a small "blob" of data stored on the client machine during web browsing in order to maintain state [Kristol, 1997]. Cookies can be temporary (that is, they are destroyed when the browser session exits) or they can be permanent – that is, they persist for a specified unit of time, possibly indefinitely. Cookies are not universally negative – they are a necessary part of working with the WWW. However, cookies can be used to profile a particular user or computer across multiple web sites. This problem is far from new [Mayer, 1997], but seems to be increasing both in prominence and application as users become more aware of the issues.

WEB USAGE – PRIVACY POLICIES

One serious issue regarding use of the World Wide Web is that a user will often voluntarily disclose information about himself assuming that that information will not compromise his privacy. Users type personal information into a competition or survey without reading the electronic small print – that is, the print that tells them that their data submission is often sold to third parties for the undisclosed or vague purposes. Similarly, some legitimate e-commerce transactions are not 100% benign. Several well-known web sites enhance revenues by selling private information (such as name and address, buying profile, and email address). This fact is disclosed on publicly available Web site privacy policies.

EMAIL – SPAM TRACKING PIXELS

The advent of HTML-enabled email has caused several issues for those concerned with privacy. In certain popular email clients (such as Outlook), emails can be previewed in a preview pane. In the case of an HTML email, however, this preview can show whether the email was opened, indicating to the sender that the email address is "live." Some spammers will attempt to send email to "predictable" email addresses at domains and use tracking pixels to ascertain opens. These addresses of these "opened" emails are deemed more valuable; in essence, the spammer knows a live address has been found and that the message was read.

DOWNLOADS – END USER LICENSE AGREEMENTS

Another extremely serious issue for users is that of the End User License Agreement (EULA). When downloading software from the Internet, users often do not read the EULA. However, the EULA can contain information that is vital for interpreting the impact of the information provided upon the privacy of the user. Additionally, there are several examples of Adware (software that displays ads on the user's machine randomly, or that target ads based upon user profile) that is "piggybacked" with other, useful applications. One controversial piece of adware – and certainly one of the most well known – is the Gator Advertising Information Network (GAIN). This software provides several useful functions – and also can gather information about surfing habits etc. Gator is given as an example, however, because the EULA and privacy policy are exemplary; anyone running the current version Gator has, at some point, been given the opportunity to read the EULA and privacy policy, in which the functionality of the software is clearly described. Thus, the software discloses its behavior and operates with the users permission, yet some users complain vehemently about the software once they become aware of its operation and perceived impact on privacy.

Malicious Disclosure

As we have seen above, there are many cases of inadvertent information disclosure that are not in line with the traditional concept of malicious in nature. However, sometimes there is another avenue through which privacy is compromised: intentionally forced disclosure facilitated by Malicious Code. The current status of the Internet provides the perfect environment for Malicious Code; self-replicating code can take advantage of the high degree of homogeneity and interconnectivity of the Internet, and Trojan Horses can be easily and rapidly disseminated via the network. Furthermore, the blurred lines between data and code further increase the opportunity for the execution of rogue code.

PASSWORD-STEALING TROJANS

The concept behind a password-stealing Trojan is far from new: the idea of using a "trojanized" piece of software to grab passwords as they fly by, either directly from the keyboard or in transit over the network has been implemented many times on a raft of different platforms. There are currently many different password-stealing Trojans deployed on unsuspecting users' machines.

SPYWARE

As the Internet develops, the value of gathering data on groups of users and individual users behavior for commercial purposes increases. Thus, there is a legitimate desire for online marketers and web site creators to tailor content and offers to users for the purposes of cross-selling and up-selling, as well as lead generation. However, unlike its legitimate cousin, Adware, Spyware does not request permission from the user prior to installation; thus, a computer can silently track personally-identifiable information, and use this to modify content.

REMOTE ACCESS TROJANS

A Remote Access Trojan is a computer program that lets a user (or users) access machine resources remotely. Here, as is often the case when considering non-viral malware, the classification of such programs as Trojans depends significantly upon one's point of view: the tool in the hands of an administrator could be a useful method of remote management. In the hands of a hacker the same tool, silently allowing an intruder into one's machine, is certainly a Trojan Horse. A good example of this dilemma is the Cult of the Dead Cow's Back Orifice. This tool is a powerful and unobtrusive architecture for remote management... yet many users consider it to be a Trojan Horse. While the position is arguable (for a counterpoint, see [CDC, 2003]), from the perspective of a user who has had BO2K installed without his permission on his machine, it certainly fulfills the requirements of a Trojan Horse.

COMPUTER VIRUSES

Previously, the primary danger of computer viruses was data modification or destruction. However, with email now commonplace on the desktop, and connectivity readily available via a standard set of system calls, the ability for viruses to export confidential data is becoming problematic. For example [Symantec, 2002], to conform with APA style.

BLENDED THREATS

Blended threats combine the characteristics of viruses, worms, Trojan Horses, and Malicious Code with server and Internet vulnerabilities to initiate, transmit, and spread an attack. By using multiple methods and techniques, blended threats can rapidly spread and cause widespread damage; just as in the case of viruses, such damage is not limited to simple damage, but can involve the dissemination of private information or the installation of other threats to privacy such as Remote Access Trojans or Password-stealing Trojans.

Technical Responses to Privacy Threats

Perhaps one of the most interesting aspects of the problems outlined above is that in each case, significant reduction of risk can be achieved by modification of user behavior. In the case of inadvertent compromise, a higher awareness and more active participation in control of user information can reduce disclosure, or at least control it.

In terms of browsing the Internet, there are many controls and configuration settings with web browsers that help facilitate privacy. For example, the Platform for Privacy Preferences Project, P3P) developed by the World Wide Web Consortium (W3C) provides for the creation for machine-readable privacy policies [Marchiori, 2002]. Such policies can be read by browsers, and acted upon accordingly. Microsoft's Internet Explorer 6.0 has added support for P3P policies for cookie control, allowing cookies to be accepted or rejected based upon the user's privacy preferences [Microsoft, 2001]. Software exists which can be configured to periodically delete unwanted cookies. However, user understanding and web site support for P3P is currently sketchy at best.

Poor acceptance of technologies addressing privacy concerns is a serious problem for those tasked with maintaining large numbers of computers, and enforcing departmental or corporate policies (see survey data below). Fortunately, there are technological solutions available that allow policy to be enforced company-wide; for example, Symantec's Enterprise Security Manager is capable of enforcing rule sets for large numbers of computers automatically. Despite this technological salve, it seems that there is a significant disconnect between expressed concern and action; even informed users seem to express concern but do not follow up with actions. Similarly, protection from unwanted but legitimate software functionality is provided by inspecting most EULAs and Privacy Policies – extension of P3P to create machine-readable EULAs and policies would help automate users privacy concerns. However, until such a system is produced, reading the EULA should provide sufficient protection.

In terms of malicious privacy compromise, the solution set is yet clearer: anti-virus software protects users from the vast majority of threats. For those concerned about spyware threat mitigation is available to the user...if they choose to apply it. This point, however, is the crux of the matter, and the primary driver behind this research: do people care about their privacy, and if so, how is this reflected (or not) in their actions.

Study Goals

As outlined above, there exist many different threats to user privacy online, ranging from tracking user actions to completely taking over their machines. However, in each case the main concern is related to user behavior not technology: often a robust technological solution exists, but the crucial element is user comprehension and action.

The goal of this study, therefore, was to determine if there was consistency between a stated desire for privacy and the day-to-day actions of information security professionals related to privacy-enhancing behaviors. The hypothesis is that security practitioners believe privacy is important and they consistently practice behaviors that are consistent with their beliefs. The null hypothesis is that security practitioners believe privacy is important but their actions are not reconciled with their beliefs. If this null hypothesis is true, then the privacy they say they believe is of value is at risk. These risks are facilitated by, but are not limited to, the behaviors measured in the study.

Methodology

The preliminary design of the survey involved querying a focus group of 67 individuals working in the computer security field. In order to measure whether or not the participants valued "privacy," and to ascertain their behaviors related to certain aspects of privacy, the subjects were asked eight True/False questions related to familiarity with Personal Privacy Policy (P3P) and reading of privacy policies (their own organization, and that of sites they visited). In order to lessen possible bias with subjects determining the questions were specifically related to privacy, the question designed to assess their attitude toward the study's operational definition of privacy was placed as the 8th question, at the end of survey.

In initial findings, it was observed that no subjects expressed a familiarity with P3P; however, when queried directly using the words "personal privacy policy," a few expressed some familiarity. Thus, the survey was revised such that it was administered using the words "personal privacy policy" rather than the acronym "P3P." Several other issues were then added to measure compliance with other privacy-enhancing behaviors, such as encrypting sensitive e-mails and deleting unwanted cookies.

The final revised survey consisted of eight True/False questions designed to measure two things: six functional/operational behaviors and the subject's desire to control of information about self and transaction. It was administered to randomly selected subjects from attendees at three IT/Security Conferences held in the United States, The United Kingdom and the EU.

Responses Summary

The responses gathered in terms of True/False answers are shown in the following table, keyed by the response of the primary question concerning privacy. This was:

I like to control the disclosure of information about myself and/or my transactions.

In the US study, there were a total of 63 respondents; the UK study contained 58; the EU study contained 23. Note that despite the small number of responses in the EU study is still statistically meaningful, given that the respondent number represented over 90% of the target group. The data collected is shown below.

Question	Group	US True	US False	UK True	UK False	EU True	EU False
I am familiar with my browser P3P	Important	27	36	30	28	17	6
	Unimportant	2	6	2	0	0	0
I always encrypt sensitive email messages	Important	26	37	21	37	8	15
	Unimportant	4	4	1	1	0	0
I encrypt all emails	Important	0	63	3	55	0	23
	Unimportant	0	8	0	2	0	0
I always delete cookies I do not need	Important	39	24	30	28	13	10
	Unimportant	2	6	1	1	0	0
I always read the privacy policy of web sites I visit	Important	3	60	11	47	4	19
	Unimportant	0	8	1	1	0	0
I always read the entire EULA of new software before agreeing to install it on my computer	Important	10	53	5	53	1	22
	Unimportant	2	6	0	2	0	0
I always encrypt data on my hard disk	Important	10	53	10	48	1	22
	Unimportant	0	8	1	1	0	0

Table 1: *Aggregate data from the study for US, UK and EU audiences. Note the large disparity between concern about privacy and actual behavior.*

> Analysis

Analysis of the data is fairly straightforward, as the results are incredibly clear: even a "by eye" analysis shows that there is a huge disconnect between belief and action. In the case of each country set, the vast majority of users expressed concern over personal information disclosure. However, the actions taken (or more frequently) not taken show a massive disregard for these concerns.

It is not possible to attribute this disconnect to technological naiveté. Consider the question regarding web site privacy policies. In this case, the US data shows that of the 63 users who expressed that they valued privacy, only three always read privacy policies on Web sites. Similarly, on the question regarding End User License Agreements, only 10 users claimed to reliably read the policy. The data from the UK and EU studies show similar behavioral biases. Given the user demographics (those people attending a security conference or trade show) it is difficult to argue that users were ignorant of the dangers inherent in installing and running executable code, yet the overwhelming majority of users did not even perform the rudimentary step of checking the claims of the software supplier.

Even in cases where there is a good and free technology solution available, such as P3P, our initial data showed that while users claimed that they were aware of the technology, further questioning revealed that there was a very low understanding of this technology. While approximately 50% of respondents stated they were familiar, conversational evidence clearly indicated that this number was higher than the real statistics. Thus, many users are actually unaware of the free and embedded technology solutions available to them.

> Cognition

Given the rather surprising nature of the results and the large disconnect between belief and response, it behooves us to discuss the underlying mechanism for this situation. Although further research is indicated, it appears unlikely that unwieldiness of technical solutions can be entirely blamed for the observed data. Even in cases where technology has been introduced to safeguard user privacy, there seems to be an apathy regarding its use or even understanding.

One model for representing contradictory cognitions is the cognitive dissonance model [Festinger 1957]. This model applies when one holds two competing thoughts or actions. For example, imagine someone has just purchased a new cellular telephone phone with free WWW access, and signed a two-year service contract[2]. The next day, a new offer arrives - upgraded phone (i.e. camera phone), and free service for six months, with no contract. The person now has two competing thoughts: the belief that they signed up for a good deal, contrasted with new parameters that are, on the surface, more attractive.

The conflict, or dissonance, could be resolved in a number of different ways. The buyer could focus on the good things they got in their deal – the strengths of the offer they accepted (i.e. free WWW access, stability of two-year with no price change, etc.). They may focus on the fact it was the "right time" to make such a purchase. At the same time, they may diminish the value of the competing belief by dismissing the extra functionality (camera) as superfluous. The amount of dissonance is affected by two factors: the number of beliefs in conflict, and the importance, or strength, of those beliefs[3].

The data gathered in this current study indicate the presence of some type of dissonance between the desire to control disclosure and the thinking regarding the actual behaviors engaged in. This process certainly threatens the privacy of users, and, as most of the individuals involved in the study were decision makers or actors in the security process, has the potential for a more widespread impact. Future research will examine ways in which dissonance can be resolved in which help, rather than harm, organizational security.

> Conclusion

The results of this study provide interesting food for thought. Despite the fact that there exist many impediments to online privacy and that educated users expressed a strong concern for their privacy, the behaviors claimed by respondents do not reflect these concerns. This result is of little surprise to the security consultant, but may be of some surprise to industry observers: there is a disconnection between the risk and the behavior.

The significance of this result for future work is clear: more research should be done to understand why the behavior does not match the concern regarding privacy. As discussed above, when the human mind encounters data that is inconsistent with behavior this dissonance must be resolved. By understanding the ways in which users are currently resolving this dissonance while continuing to engage in "at risk" behaviors, education and product design can be modified such that the risk is mitigated most effectively. The weakest link in the computer security chain remains the person using the computer: research that emphasizes strengthening this crucial link will provide the largest increase of security and the best possible research benefit.

[2] People tend avoid input that will increase dissonance, however, sometimes the beliefs are forced upon them.
[3] One seemingly contradictory result noted by Festinger was that when a person acts against their internal beliefs, the smaller the reward for doing so, the larger the generated dissonance. In a classic experiment [Festinger & Carlsmith, 1959], Festinger "rewarded" participants for espousing a position that they did not actually believe. Interestingly, those who were rewarded least showed the greatest shift in their own personal belief system.

References

Al-Sabt. M. 1995. Arabian Business and Cultural Guide. Published by Traderscity.com.

Allen, A. 2000. Women, Privacy and Cyberspace, Stanford University Symposium on "Cyberspace and Privacy: A New Legal Paradigm".

CDC. 2003. Cult of the Dead Cow, A Note on Product Legitimacy and Security, available online at http://www.bo2k.com/docs/bo2k_legitimacy.html

Chess, D. 2003. Personal correspondence. Used with permission.

Festinger, L. 1957. Theory of Cognitive Dissonance. Stanford University Press. Stanford, CA:

Festinger, L. & Carlsmith, J. 1959. Cognitive Consequences of Forced Compliance. Journal of Abnormal and Social Psychology, 58, pp. 203-210.

Fullbright. 2003. Retrieved from the World Wide Web July 1, 2003. http://www.fulbright.jp/e4/ayjcont4.html.

IBM Corporation. 1999. IBM Multi-National Consumer Privacy Study. IBM Global Services. IBM Corporation. Yorktown Heights, NY.

ITAA. 2003. Security and Privacy National Attitudes Research Study. Retrieved from the World Wide Web on July 15, 2003. http://www.itaa.org/infosec/faith.pdf

Kaminsky, J. 2003. Personal Correspondence. Used with permission.

Kristol D. & Montulli L. 1997. HTTP State Management Mechanism, RFC2169 1997

Marchiori, M. 2002. The Platform for Privacy Preferences 1.0 (P3P 1.0) Specification. Marchiori M., Ed. Available online at http://www.w3.org/TR/P3P/ (2002).

Marx, G. 2003. Technology and Gender: Thomas I. Voire and the Case of the Peeping Tom. The Sociological Quarterly, Volume 43, Number 3, pp. 407-433.

Mayer-Schonberger, V. 1997. The Internet and Privacy Legislation: Cookies for a Treat?, 1 West Virginia Journal of Law & Technology 1,1.

Microsoft, 2001. Internet Explorer 6 Technical Overview. Retrieved from the World Wide Web July, 2003. http://www.microsoft.com/windows/ie/techinfo/overview/default.asp. Posted October 08, 2001.

OED. 2003. Oxford Dictionary. Oxford University Press.

Raiu, C. 2003. Personal Correspondence. Used with permission.

Shipp, A. 2003. Personal Correspondence. Used with permission.

Stefnisson, S. 2003. Personal Correspondence. Used with permission.

Symantec, 2002. Symantec Security Response. Analysis of W32.Bugbear@mm. Available online at http://www.symantec.com/avcenter/venc/data/w32.bugbear@mm.html

Websters, 2003. Websters Unabridged Dictionary.

Whalley, I. 2003. Personal Correspondence. Used with permission.

About the Author

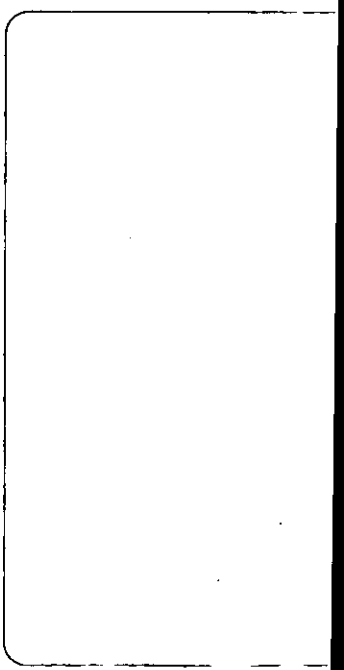
Sarah Gordon is Senior Research Fellow at Symantec Security Response. Her current research areas include testing and standards for antivirus and security software, privacy issues, cyberterrorism and psychological aspects of human/computer interaction.

The has been featured in diverse publications such as IEEE Monitor,The Wall Street Journal, and Time Digital, and profiled by PBS, ITN, and CNN International. Her work has appeared in publications such as Information Security News and Virus Bulletin; she has won several awards for her work in technology. She is a highly sought-after speaker, having presented at conferences ranging from DEFCON to Govsec.

She was recently appointed to the Editorial Board for Elsevier Science Computers and Security Journal. She is on the Advisory Board of Virus Bulletin, and is co-founder and board member of The WildList Organization International. She is Technical Director of The European Institute for Computer Antivirus Research - where she also serves on the Board of Directors and Conference Program Committee. She is a member of SRI's cyberadversary working group.

Sarah was responsible for security testing and recommendation for The United Nations, and participates in various initiatives for Homeland Security and Infrastructure Protection. She was chosen to represent the security industry in "Facts on File: Careers for Kids who Like Adventure", and is a reviewer for various technical security book publishers including Wiley publications. Her work in ethics, technology and profiling computer criminals is required coursework in various academic information security programs. She is committed to excellence in information security education, guest lecturing at Universities world-wide on topics ranging from virus writers and hackers to the truth about cyberterrorism.

Sarah graduated from Indiana University with special projects in both UNIX system security and ethical issues in technology. She is a member of the American Association for the Advancement of Science, The American Counseling Association, and the Association for Family Therapy and Systemic Practice in the UK. Prior to joining Symantec, she worked with the Massively Distributed Systems Group at IBM's Thomas J. Watson Research Laboratory in New York in the AntiVirus Research and Development Team. She may be reached at sgordon@symantec.com.



SCAN COVER